

Dr. Dobb's Journal of Software Tools

FOR THE PROFESSIONAL PROGRAMMER

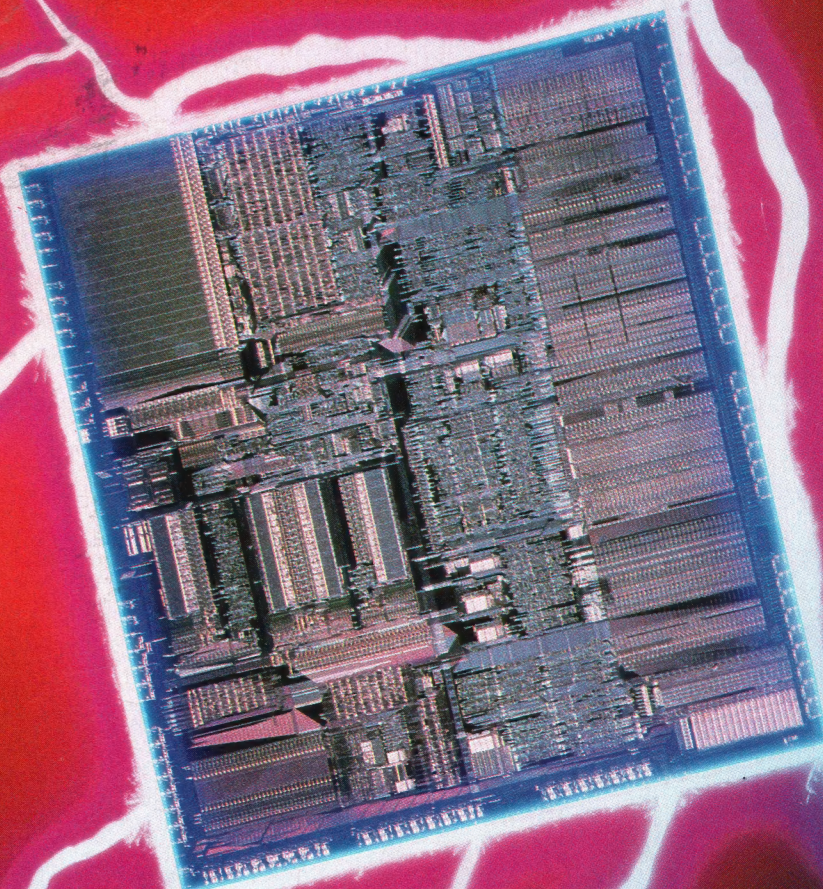
80386 PROGRAMMING

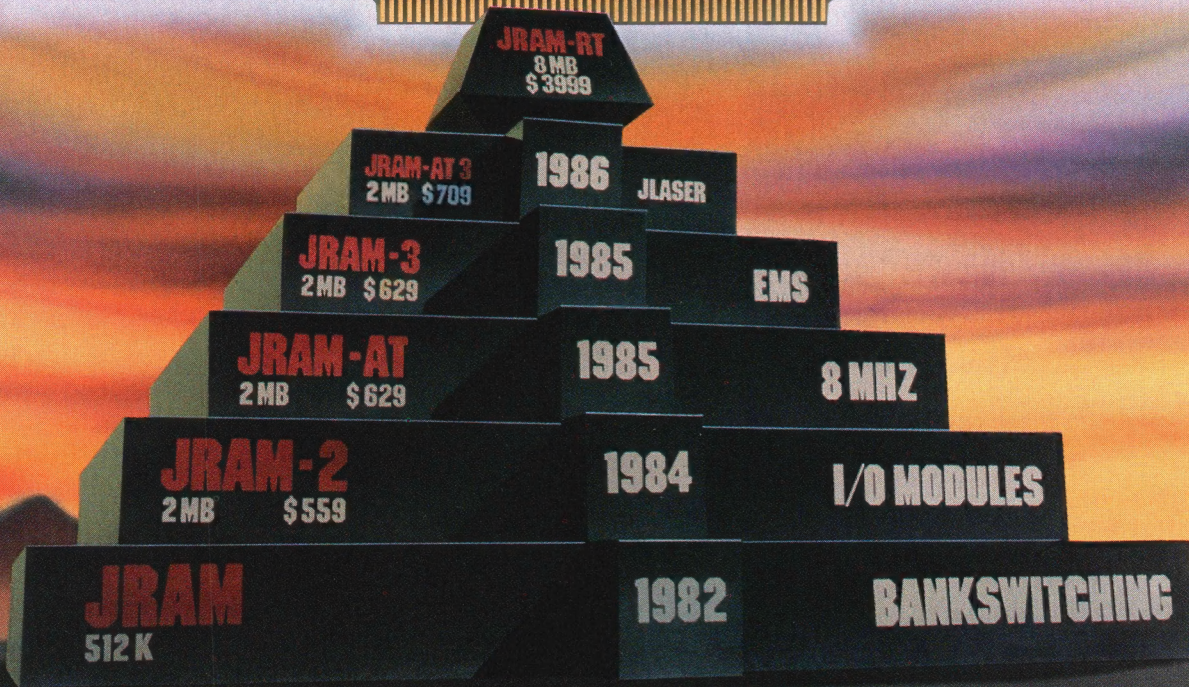
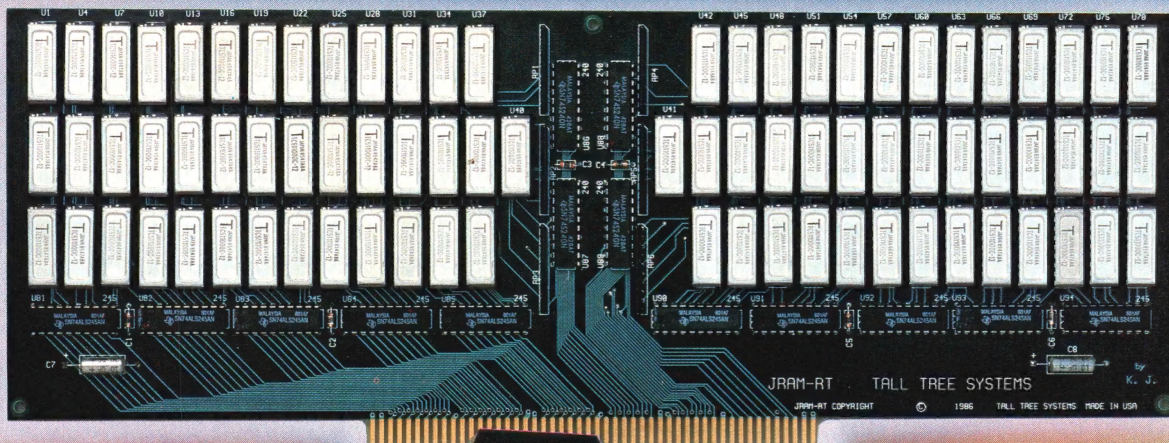
MS-DOS
File Browsing

Converting
to the 320xx

Modula-2
Compiler Review

Factoring
in Forth





TALL TREE SYSTEMS. A Technological Innovator. Always a Step Ahead!

For true industry leadership, look no further than Tall Tree Systems.

We have a history of being first.

We were the first to introduce bankswitching. The first with two megabyte memory boards. The first with I/O modularity in a single slot. The first with 8 MHz speed capabilities. The only maker of single

command EMS boards.

The first with a laser printer solution — JLASER — that allows you to do full-page graphics and multiple type fonts on any Canon® or Ricoh® laser engine.

Now, we're first again with memory expansion for the IBM® RT.

Innovation is our tradition. Our trademark is superior technology at the lowest possible price.



TALL TREE SYSTEMS

1120 San Antonio Road • Palo Alto, CA 94303 • (415) 964-1980

© 1986 by Tall Tree Systems. All rights reserved. IBM, RT are registered trademarks of International Business Machines Corp. Canon and Ricoh are registered trademarks of Canon Corp. and Ricoh Corp., respectively.

Circle no. 185 on reader service card.

MARK WILLIAMS C. AN ENLIGHTENING DEVELOPMENT FOR ATARI ST USERS.

If you've tried your hand at developing applications on the Atari ST, you know the problem. Programming tools aren't only hard to come by, they're hard to use. One might even say primitive. But now for some enlightening news: you can have all the power, portability and versatility of the C language from a leader in professional C programming tools, Mark Williams.

BRING YOUR PROGRAMMING UP TO SPEED.

The Mark Williams C compiler produces fast, dense code and supports the complete Kernighan & Ritchie industry standard C. You'll have access to GEM's AES and VDI libraries for programs using graphics, icons and the Atari mouse. And Mark Williams C lets you take advantage of the full 16 megabytes in Atari's 68000 microprocessor.

STREAMLINE DEVELOPMENT WITH POWER UTILITIES.

Mark Williams C is loaded with everything you'll need for professional development. Bring the power of the UNIX environment to your Atari ST with our

Features

C compiler

- Complete Kernighan & Ritchie C plus extensions
- Up to eight register variables
- Full access to AES and VDI libraries for programs using graphics, icons and mouse
- Complete UNIX-compatible libraries allow easy portability to and from UNIX development environment.
- Over 300 Atari-specific routines
- One-step compiling, linking with cc command
- English error messages
- Lint-like error checking

Microshell Command Processor, powerful UNIX style shell includes I/O redirection, pipes, command substitutions

MicroEMACS Full Screen Editor with commented source code included

Make Program Building Discipline

Complete symbolic debugger with single-step, breakpoints and stack traceback

Assembler, linker and archiver
Powerful Utilities Package: egrep, sort, diff, cmp, pr, tail, uniq, wc and more

Over 600 pages of documentation including 120 sample C programs
Not copy protected

MARK WILLIAMS C FOR THE ATARI ST
\$179.95
60 DAY MONEY BACK GUARANTEE



Microshell Command Processor including pipes, I/O redirection and more. Edit your program with the highly acclaimed MicroEMACS full screen editor. Accelerate and simplify compiling with *make* which finds and recompiles only those modules affected by your changes. Then, when you're ready for debugging, call on our db Symbolic Debugger with single step, breakpoint and stack traceback functions. Over 40 commands, including a linker and assembler, provide a total development package for your Atari ST.

DEPEND ON A NAME WITH A HISTORY OF PERFORMANCE.

Mark Williams C for the Atari ST is part of our growing line of C compilers. A line that includes the C compiler chosen by DEC, Intel, Wang and thousands of professional programmers. Now our Atari C compiler is earning its own reputation:

"Finally a great C compiler that exploits the power of the ST"—Sigmund Hartmann, President, Atari Software Group

"The all-around best choice for serious software development on the ST."—Douglas Weir of *ANALOG COMPUTING*

GET WHAT YOUR ATARI ST HAS BEEN WAITING FOR.

Mark Williams C is just what your Atari ST was made for: powerful, professional programming. So now that you can have Mark Williams C for just \$179.95, what are you waiting for?

Ask your Atari dealer about Mark Williams C or order today by calling **1-800-MWC-1700.***

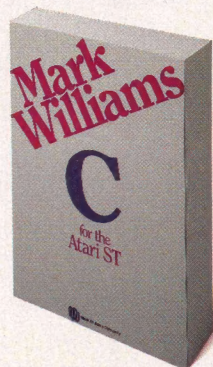
*In Illinois call: 312-472-6659

 **Mark
Williams
Company**

1430 West Wrightwood, Chicago, Illinois 60614

© 1986, Mark Williams Company
UNIX is a trademark of Bell Labs.

Circle no. 102 on reader service card.



Instant Replay

NEW

Synthetic Intelligence
Patent Pending

Generates:

DEMOS, TUTORIALS,
PROTOTYPES, PRESENTATIONS,
TIMED KEYBOARD MACROS, AND
MENU SYSTEMS

Instant Replay memorizes how you ran your program and Instantly generates a Demo Replay. Keystrokes and pause times, inserted prompts, pop-ups and prototypes are all memorized.

Screen Painter for creating 100% user designable pop-ups and Menu Windows. "Excellent, intuitive, easiest-to-use screen generator reviewed" PC Magazine .

Run-Time Screen Scanner and File Scanner for creating Menu systems and presentations.

Screen Grabber so you can edit and include any screen.

Text Editor - fast, full function, with pull-down menus that can be tailored. Memorize it and distribute it as a Demo!

200 Page Manual, 4 diskettes, 60 Day money back guarantee.
(Not Copy Protected)

Call or Write. We accept Visa, Amex, Master Card, COD, PO.
Dealer Inquiries Welcome.

Instant Replay ^(tm)	\$ 89.95
Demo Diskette	\$ 5.00
Dealer Poster	\$ 3.00

Nostradamus

Nostradamus Inc. / 5320 South 900 East, Suite 110
SLC, Utah 84117 / Order by phone (801) 261-0769

For IBM PC / XT / AT and True Compatibles

Ask about Programming Libraries

Circle no. 251 on reader service card.

Dr. Dobb's Journal of Software Tools

FOR THE PROFESSIONAL PROGRAMMER

ARTICLES

**80386
programming** ▶

PROCESSORS: Programming on the 80386 **28**
by Ross Nelson
Ross discusses native-mode operation and performance considerations for the 32-bit microprocessor.

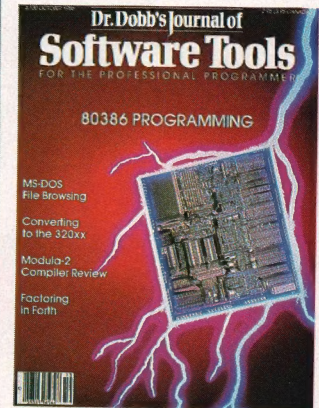
**Converting to
the NS320xx** ▶

PROCESSORS: TNZ: An 8-bit to 16-bit Translator **40**
by Richard A. Campbell
How to convert programs written for the Z80 to run on the NS 320xx processors.

**Modula-2
compilers
compared** ▶

REVIEWS

LANGUAGES: Modula-2 Compilers for the IBM PC **48**
by Namir Clement Shamma
A comparative look at four of the latest compilers. Benchmark results are included.



About the Cover

Art director Michael Hollister and Michael Carr/Pacific Horizons enjoyed shooting the photogenic Intel 80386.

**File Browsing
in MS-DOS** ▶

COLUMNS

C CHEST: More, a File-Browsing Utility **22**
by Allen Holub

Allen shows how to page through a file without bothering with an editor.

**MS-DOS hints
and resources** ▶

16-BIT SOFTWARE TOOLBOX: MS-DOS Tricks **96**
by Ray Duncan

Ray and his readers present a variety of helpful hints and more resources for MS-DOS programmers. Ray uncovers an ingenious text-searching algorithm that works in a mysterious manner.

**Forth
techniques** ▶

STRUCTURED PROGRAMMING: Factoring in Forth **104**
by Michael Ham

Michael follows factoring throughout Forth, showing how good factoring skills are applicable to every level of the language.

This Issue

In our feature article, we take a close look at Intel's 80386 chip. What does it really offer, and how do you upgrade from the 80286? Richard A. Campbell shows how to convert programs written for the 8-bit Z80 microprocessor to the 16-bit NS320xx chip set. Our review this month is a comparison of four Modula-2 compilers from the latest generation. Forth factoring is Michael Ham's topic in Structured Programming, and Allen Holub presents a file-browsing utility in C Chest.

Next Issue

New graphics controllers have provided software developers with increased capabilities—and new complexities. In our graphics issue, Ed McNierney discusses the issues these controllers raise, the opportunities they present, and new programming techniques to help programmers realize the potential of the chips.

FORUM

EDITORIAL **6**
by Michael Swaine
RUNNING LIGHT **8**
by Nick Turner
ARCHIVES **8**
LETTERS **10**
by you
SWAINE'S FLAMES **136**
by Michael Swaine

PROGRAMMER'S SERVICES

DR. DOBB'S CATALOG: **105**
DDJ products—all in one place
THE PROFESSIONAL PROGRAMMER: **126**
Professional organizations for programmers
OF INTEREST: Many **128**
new products of interest to programmers
ADVERTISER INDEX: **134**
Where to find those ads

**Why are you
having fun?** ▶

**Which is to be
master?** ▶

**YOUR
COMPUTER LANGUAGE
IS QUIETLY
BREEDING REAL BATS
IN YOUR
BELFRY.**



LANGUAGES THAT ARE CAUSING THE BIGGEST PROGRAMMING BACKLOG IN HISTORY ARE ALSO EATING NICE BIG HOLES IN OUR POCKETS.

Whether it's BASIC, COBOL, Pascal, "C", or a data base manager, you're being held back.

Held back because the language has frustrating limitations, and the programming environment isn't intuitive enough to keep track of what you're working on.

In the real world, there's pressure to do more impressive work, in less time, and for more clients.

We've been given some incredibly powerful hardware in recent times, but the languages aren't a whole lot better than they were 20 years ago.

So, whatever language you have chosen, by now you feel it's out to get you — because it is.

Sure, no language is perfect, but you have to wonder, "Am I getting all I deserve?"

And, like money, you'll never have enough.

Pretty dismal, huh?

We thought so, too.

So we did something about it.

We call it CLARION™.

You'll call it "incredible."

Distributed on 7 diskettes, CLARION consists of over 200,000 lines of code, taking 3+ years to hone to "world-class" performance.

With CLARION you can write, compile, run and debug complex applications in a New York afternoon.

Even if you're in Savannah.

It gives you the power and speed to create screens, windows and reports of such richness and clarity you would never attempt them with any other language.

Because you would have to write the code.

With CLARION you simply design the screens using our SCREENER utility and then CLARION writes the source code AND compiles it for you in seconds.

Likewise, you can use REPORTER to create reports.

Remember, only CLARION can recompile and display a screen or report layout for modification.

And with no time wasted.

All the power and facilities you need to write great programs, faster than you ever dreamed of.

Programs that are easy to use.
Programs that are a pleasure to write.

And to you that means true satisfaction.

You've coveted those nifty pop-up help windows some major applications feature. But you can't afford the time and energy it takes to write them into your programs.

That's the way it used to be.

So we fixed that, too.

CLARION's HELPER is an interactive utility that let's you design the most effective pop-up help screens that you can imagine. And they're "context sensitive," meaning you can have help for every field in your application.

Unlike the other micro languages, CLARION provides declarations, procedures, and functions to process dates, strings, screens, reports, indexed files, DOS files and memory tables.

Imagine making source program changes with the CLARION EDITOR. A single keystroke terminates the EDITOR, loads the COMPILER, compiles the program, loads the PROCESSOR and executes the program. It's that easy!

Our data management capabilities are phenomenal. CLARION files permit any number of composite keys which are updated dynamically.

A file may have as many keys as it needs. Each key may be composed of any fields in any order. And key files are updated whenever the value of the key changes.

Like SCREENER and REPORTER, CLARION's FILER utility also has a piece of the CLARION COMPILER. To create a new file, you name the Source Module. Then you name the Statement Label of a file structure within it.

FILER will also automatically rebuild existing files to match a changed file structure. It creates a new record for every existing record, copying the existing fields and initializing new ones.

Sounds pretty complicated, huh?

Not with CLARION's documentation and on-line help screens. If you are currently competent in BASIC, Pascal or "C" you can be writing CLARION applications in a day. In two days you won't believe the eloquence of your CLARION programs.

Okay, now for the best part of all. You can say it in CLARION for \$295.00—plus shipping and handling. All you need is an IBM® PC, XT, AT or true compatible, with 320 KB of memory, a hard disk drive, and a parallel port. And we'll allow a full 30 day evaluation period. If you're not satisfied with

CLARION, simply return it in its original condition for a full refund.

If you're not quite ready to take advantage of this no-risk opportunity, ask for our detailed 16 page color brochure. It vividly illustrates the elegance of CLARION. Consider it a preview of programming in the fast lane.

Either way, the 800 call's a freebie.



SAY IT IN

CLARION™

A4ST/5

1-800-354-5444

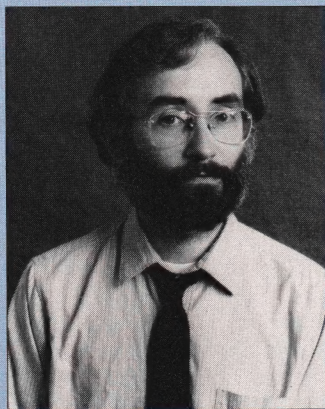


BARRINGTON SYSTEMS, INC. 150 EAST SAMPLE ROAD POMPANO BEACH, FLORIDA 33064 305/785-4555

IBM is a registered trademark of International Business Machines Corporation. CLARION™ is a trademark of Barrington Systems, Inc. ©1986 Barrington Systems

Circle no. 115 on reader service card.

EDITORIAL



Warning: the following editorial contains many personal pronouns used in a way that permits people to be considered as abstractions and men to be confused with women. So far as I have been able to determine, this usage does not violate any of the guidelines set forth by the Meese commission on pornography, but, just to be on the safe side, anyone likely to be offended by such ideas is cautioned not to read further.

Something has been puzzling me about you. I mean the plural you, the statistical you. After having hundreds of conversations at shows and on the phone and reading letters and reader-survey data, I've chewed up all the scraps of information I could get and pasted them into this papier-mâché person, this useful abstraction, the *Dr. Dobb's* reader.

DDJ readers are not dilettantes, not hobbyists playing with the technology. You are professionals, and your knowledge pays your rent. Something about this has always puzzled me: if you're such serious professionals, why are you having so much fun?

The answer, I have decided, has to do with levels of programming knowledge. The first level is occupied by the programmer with raw skill. You know her: she can always shave off another machine cycle or squeeze out another byte. Give her a dimension and she'll optimize along it. Just be sure to tell her what dimension is important, or she'll give you small when you need fast or fast when you need small. Programming to her is like juggling or puzzle solving—always a challenge, always fun. That spirit was involved in the founding of this magazine, and I hope something of it still persists.

At a level above the enthusiast is the professional. I don't mean that

the professional knows more or is a better programmer; the difference is that the professional augments her raw skills with another level of knowledge—judgment about how to apply those skills. For the professional, the task is not always fun or chal-

lenging, however much it may challenge her design skills or her task-management abilities. Sometimes what's required isn't dazzle but drudge work.

The *DDJ* readers I talk with at shows are professionals, but they—you—always seem to be solving interesting problems. You don't seem bored. You don't seem to be doing any drudge work. Why is that?

I finally figured it out, and the answer is something I'm sure you already know: there is a level of knowledge beyond professionalism. It was right there in the reader-survey data. Some of you run your own companies. Others head design teams. Many of you are simply in a position to call the shots, to pick your own projects.

Just as the knowledge of how to apply raw skill separates the professional from the enthusiast, the freedom to decide which problems to pursue distinguishes you from the professional. Just as it's assumed that the professional has the necessary programming skills for the job, it's assumed that you have the professional knowledge to decide where to apply your programming skills.

Because you can choose the tasks, you pick tasks you like. You can decide whether a project will be challenging or enjoyable enough for you.

You lucky dog, you.

Michael Swaine

Michael Swaine
editor-in-chief

Dr. Dobb's Journal of Software Tools

Editorial

Editor-in-Chief Michael Swaine
Editor Nick Turner
Managing Editor Vince Leone
Assistant Editor Sara Noah Ruddy
Technical Editor Allen Holub
Contributing Editors Ray Duncan
Michael Ham
Allen Holub
Namir Shamas

Copy Editor Rhoda Simmons
Electronic Editor Levi Thomas

Production

Production Manager Bob Wynne
Art Director Michael Hollister
Assoc. Art Director Alida Hinton
Typesetter Jean Aring
Cover Artist Michael Carr

Circulation

Newsstand Sales Mgr. Stephanie Barber
Circulation Director Maureen Kaminski
Book Marketing Mgr. Jane Sharninghouse
Circulation Assistant Kathleen Shay

Administration

Finance Manager Treva Rafalski
Business Manager Betty Trickett
Accounts Payable Supv. Mayda Lopez-Quintana
Accts. Pay. Coordinator Kathy Robinson
Accounts Payable Asst. Karen Green
Accounts Receivable Mgr. Laura Di Lazzaro
Accts. Receivable Asst. Denise Giannini
Adm. Coordinator Kobi Morgan
Advertising Director
Robert Horton (415) 366-3600
Account Managers
Michele Beaty (317) 875-8093
Lisa Boudreau (415) 366-3600
Gary George (404) 897-1923
Michael Wiener (415) 366-3600
Cynthia Zuck (718) 499-9333
Promotions/Srvcs. Mgr. Anna Kittleson
Advertising Coordinator Michelle A. Davié

M&T Publishing, Inc.

Chairman of the Board Otmar Weber
Director C.F. von Quadl
President and Publisher Laird Foshay

Dr. Dobb's Journal of Software Tools (USPS 307690) is published monthly by M&T Publishing, Inc., 501 Galveston Dr., Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points.

Article Submissions: Send manuscripts and disk (with article and listings) to the Assistant Editor.

Address Correction Requested: Postmaster: Send Form 3579 to *Dr. Dobb's Journal*, P.O. Box 27809, San Diego, CA 92128. **ISSN 0898-3078**

Customer Service: For subscription problems call: outside CA (800) 321-3333; within CA (619) 485-9623 or 566-6947. For order problems call (415) 366-3600.

Subscription Rates: \$29.97 per year, U.S. Foreign rates: \$63.97, air; \$51.97, surface. Foreign subscriptions must be prepaid in U.S. dollars, drawn on a U.S. bank. For foreign subscriptions, TELEX: 752-351.

Foreign Newsstand Distributor: Worldwide Media Service, Inc., 386 Park Ave. South, New York, NY 10016; (212) 686-1520 TELEX: 620430 (WU).

Entire contents copyright © 1986 by M&T Publishing, Inc.; unless otherwise noted on specific articles. All rights reserved.



People's Computer Company

Dr. Dobb's Journal of Software Tools is published by M&T Publishing, Inc. under license from People's Computer Company, 2682 Bishop Dr., Suite 107, San Ramon, CA 94583, a nonprofit corporation.



The C for Microcomputers

PC-DOS, MS-DOS, CP/M-86, Macintosh, Amiga, Apple II, CP/M-80, Radio Shack, Commodore, XENIX, ROM, and Cross Development systems

MS-DOS, PC-DOS, CP/M-86, XENIX, 8086/80x86 ROM

Manx Aztec C86

"A compiler that has many strengths ... quite valuable for serious work"

Computer Language review, February 1985

Great Code: Manx Aztec C86 generates fast executing compact code. The benchmark results below are from a study conducted by Manx. The Dhrystone benchmark (CACM 10/84 27:10 p1018) measures performance for a systems software instruction mix. The results are without register variables. With register variables, Manx, Microsoft, and Mark Williams run proportionately faster, Lattice and Computer Innovations show no improvement.

	Execution Time	Code Size	Compile/Link Time
Dhrystone Benchmark			
Manx Aztec C86 3.3	34 secs	5,760	93 secs
Microsoft C 3.0	34 secs	7,146	119 secs
Optimized C86 2.20J	53 secs	11,009	172 secs
Mark Williams 2.0	56 secs	12,980	113 secs
Lattice 2.14	89 secs	20,404	117 secs

Great Features: Manx Aztec C86 is bundled with a powerful array of well documented productivity tools, library routines and features.

Optimized C compiler	Symbolic Debugger
AS86 Macro Assembler	LN86 Overlay Linker
80186/80286 Support	Librarian
8087/80287 Sensing Lib	Profiler
Extensive UNIX Library	DOS, Screen, & Graphics Lib
Large Memory Model	Intel Object Option
Z (vi) Source Editor -c	CP/M-86 Library -c
ROM Support Package -c	INTEL HEX Utility -c
Library Source Code -c	Mixed memory models -c
MAKE, DIFF, and GREP -c	Source Debugger -c
One year of updates -c	CP/M-86 Library -c

Manx offers two commercial development systems, Aztec C86-c and Aztec C86-d. Items marked -c are special features of the Aztec C86-c system.

Aztec C86-c Commercial System	\$499
Aztec C86-d Developer's System	\$299
Aztec C86-p Personal System	\$199
Aztec C86-a Apprentice System	\$49

All systems are upgradable by paying the difference in price plus \$10.

Third Party Software: There are a number of high quality support packages for Manx Aztec C86 for screen management, graphics, database management, and software development.

C-tree \$395	Greenleaf \$185
PHACT \$250	PC-lint \$98
HALO \$250	Amber Windows \$59
PRE-C \$395	Windows for C \$195
WindScreen \$149	FirstTime \$295
SunScreen \$99	C Util Lib \$185
PANEL \$295	Plink-86 \$395

MACINTOSH, AMIGA, XENIX, CPM-68K, 68k ROM

Manx Aztec C68k

"Library handling is very flexible ... documentation is excellent ... the shell a pleasure to work in ... blows away the competition for pure compile speed ... an excellent effort."

Computer Language review, April 1985

Aztec C68k is the most widely used commercial C compiler for the Macintosh. Its quality, performance, and completeness place Manx Aztec C68k in a position beyond comparison. It is available in several upgradable versions.

Optimized C Macro Assembler	Creates Clickable Applications
Overlay Linker	Mouse Enhanced SHELL
Resource Compiler	Easy Access to Mac Toolbox
Debuggers	UNIX Library Functions
Librarian	Terminal Emulator (Source)
Source Editor	Clear Detailed Documentation
MacRam Disk -c	C-Stuff Library
Library Source -c	UniTools (vi, make, diff, grep) -c
	One Year of Updates -c

Items marked -c are available only in the Manx Aztec C86-c system. Other features are in both the Aztec C86-d and Aztec C86-c systems.

Aztec C68k-c Commercial System	\$499
Aztec C68d-d Developer's System	\$299
Aztec C68k-p Personal System	\$199
C-tree database (source)	\$399
AMIGA, CP/M-68k, 68k UNIX	call

Apple II, Commodore, 65xx, 65C02 ROM

Manx Aztec C65

"The AZTEC C system is one of the finest software packages I have seen"

NIBBLE review, July 1984

A vast amount of business, consumer, and educational software is implemented in Manx Aztec C65. The quality and comprehensiveness of this system is competitive with 16 bit C systems. The system includes a full optimized C compiler, 6502 assembler, linkage editor, UNIX library, screen and graphics libraries, shell, and much more. The Apple II version runs under DOS 3.3, and ProDOS, Cross versions are available.

The Aztec C65-c/128 Commodore system runs under the C128 CP/M environment and generates programs for the C64, C128, and CP/M environments. Call for prices and availability of Apprentice, Personal and Developer versions for the Commodore 64 and 128 machines.

Aztec C65-c ProDOS & DOS 3.3	\$399
Aztec C65-d Apple DOS 3.3	\$199
Aztec C65-p Apple Personal system	\$99
Aztec C65-a for learning C	\$49
Aztec C65-c/128 C64, C128, CP/M	\$399

Distribution of Manx Aztec C

In the USA, Manx Software Systems is the sole and exclusive distributor of Aztec C. Any telephone or mail order sales other than through Manx are unauthorized.

Manx Cross Development Systems

Cross developed programs are edited, compiled, assembled, and linked on one machine (the HOST) and transferred to another machine (the TARGET) for execution. This method is useful where the target machine is slower or more limited than the HOST, Manx cross compilers are used heavily to develop software for business, consumer, scientific, industrial, research, and educational applications.

HOSTS: VAX UNIX (\$3000), PDP-11 UNIX (\$2000), MS-DOS (\$750), CP/M (\$750), MACINTOSH (\$750), CP/M-68k (\$750), XENIX (\$750).

TARGETS: MS-DOS, CP/M-86, Macintosh, CP/M-68k, CP/M-80, TRS-80 3 & 4, Apple II, Commodore C64, 8086/80x86 ROM, 68xxx ROM, 8080/8085/Z80 ROM, 65xx ROM.

The first TARGET is included in the price of the HOST system. Additional TARGETS are \$300 to \$500 (non VAX) or \$1000 (VAX).

Call Manx for information on cross development to the 68000, 65816, Amiga, C128, CP/M-68K, VRTX, and others.

CP/M, Radio Shack, 8080/8085/Z80 ROM

Manx Aztec CII

"I've had a lot of experience with different C compilers, but the Aztec C80 Compiler and Professional Development System is the best I've seen."

80-Micro, December, 1984, John B. Harrell III

Aztec C II-c (CP/M & ROM)	\$349
Aztec C II-d (CP/M)	\$199
C-tree database (source)	\$399
Aztec C80-c (TRS-80 3 & 4)	\$299
Aztec C80-d (TRS-80 3 & 4)	\$199

How To Become an Aztec C User

To become an Aztec C user call 1-800-221-0440 or call 1-800-832-9273 (800-TEC WARE). In NJ or outside the USA call 201-530-7997. Orders can also be telexed to 4995812.

Payment can be by check, COD, American Express, VISA, Master Card, or Net 30 to qualified customers.

Orders can also be mailed to Manx Software Systems, Box 55, Shrewsbury, NJ 07701.

How To Get More Information

To get more information on Manx Aztec C and related products, call 1-800-221-0440, or 201-530-7997, or write to Manx Software Systems.

30 Day Guarantee

Any Manx Aztec C development system can be returned within 30 days for a refund if it fails to meet your needs. The only restrictions are that the original purchase must be directly from Manx, shipped within the USA, and the package must be in resalable condition. Returned items must be received by Manx within 30 days. A small restocking fee may be required.

Discounts

There are special discounts available to professors, students, and consultants. A discount is also available on a "trade in" basis for users of competing systems. Call for information.

MANX

To order or for information call:

800-221-0440

UNIX is a registered TM of Bell Laboratories. Lattice TM Lattice Inc. C-tree TM Faircom, Inc. PHACT TM PHACT ASSOC. C Optimizing C86 TM Computer Innovations, Inc. MACINTOSH, APPLE TM APPLE, INC. Pre-C, PLINK 86 TM PHOENIX, HALO TM Media Cybernetics, Inc. C-tree, PC-lint TM GIMPLE Software, WindScreen, SunScreen TM SunSoft, PANEL TM Roundhill Computer Systems Ltd. WINDOWS FOR C TM Creative Solutions, XENIX, MS TM MICROSOFT INC. CP/M TM DRI, AMIGA, C64, C128 TM COMMODORE Int.

Circle no. 108 on reader service card.

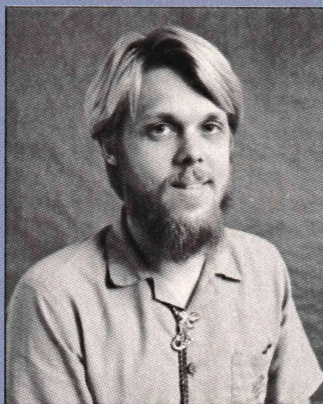
RUNNING LIGHT

It seems that we have reached the goal that the 8- and 16-bit microprocessors of the past were leading up to: true 32-bit microprocessors, designed with the programmer in mind. The new generation of 32-bit microprocessors have a lot to offer programmers. We were particularly interested in what Intel's 80386 would mean to software developers, and Ross Nelson's feature article in this issue represents the beginning of our exploration of programming on the 80386. In this and subsequent issues, we'll also be looking at the other 32-bit processors, including the National Semiconductor 32332 and the Motorola 68020.

We get a lot of letters, telephone calls, and E-mail messages from readers asking to see more articles on a specific topic. Why don't you support OS-9? When are you going to publish something on the Atari ST? How about more Unix (or FORTRAN or 80386) coverage? In fact, we have plans to cover all those topics in upcoming issues, but you should understand that how well and how frequently we cover your favorite topic depends on you: almost all our articles are produced by our readers.

There are a number of topics that we're particularly interested in. Do any of these areas match your expertise and interest? Scientific computing. Fourth-generation languages. Programming and the 80386. The 68000 machines: Macintosh, Atari ST, Commodore Amiga; and the 68020 machines, like the Mustang 020. OS-9. VersaDOS. The Hypercube. Graphics techniques. Pattern recognition. Machine learning. Do you have an idea? Give me a call.

Here's what's coming up early in 1987. In future months, I'll be talking



about some of these issues in greater detail.

March: Data compression. Ten years ago, the challenge was to pack significant processing power into a small amount of memory. Today, the equivalent challenge may be to rapidly move

masses of information over narrow channels. Article deadline: November 1, 1986.

April: Artificial intelligence. Will machine learning be the next big thing in practical AI, as AI pioneer Patrick Henry Winston believes? Deadline: December 1, 1986.

May: Arts and sciences. We'll look into computers and music, and programming for scientific applications. Deadline: January 1, 1986.

June: Our annual telecommunications issue. Deadline: February 1, 1986.

We also want to broaden our linguistic horizons in 1987, so don't be shy about sending in Pascal code, or FORTRAN, or LISP, or whatever. If you have an article idea, call me at (415) 366-3600 and we'll discuss it.

Nick Turner
editor

ARCHIVES

The Spirit of 76

"Ever since I first saw 73 magazine (a ham radio mag out of Peterborough NH) and noticed that they used the 'radical' scheme of simply numbering their issues sequentially rather than using a Volume number and Issue number, I have been thinking of switching Dr. Dobbs... to rid ourselves of this sector/byte addressing in favor of simple, linear byte addressing."—*Jim Warren, DDJ, October 1976.*

"Can you program a working tic-tac-toe game in an hour without any brainstrain? Are you looking for a way to make a living programming games and systems and exploring the strange wonders of software? We are a small engineering group inventing games... in Grass Valley, California in the Sierra foothills an hour from skiing and two hours from Chinese food."—career opportunity, DDJ, October 1976.

"Dear Dr. Dobb,

I would like to express my opinions about the two conversion formats of computer address and data information currently implemented on nearly all micro- and mini-computers. I believe hex to be dominant over octal in many aspects:

Since all computers' address and data word sizes are in multiples of four (4) bits, octal representation often wastes a digit; that digit representing one or two bits instead of three.

Hex never wastes a digit: each nibble (four bits) of a number is represented by one hex digit.

In the instance of the PDP-8, a twelve (12) bit mini-computer, one address word can be represented by four full octal digits without waste. But—only three hex digits are needed to stand for twelve bits, also without waste.

When one memorizes an instruction set, types in an object program, or prints out an assembler listing, thousands upon thousands (1000/1000) of wasted characters are spewed out, typed in, or memorized unnecessarily when the octal format is implemented instead of hex (especially the amazingly unorthodox 'split' octal is used).

In conclusion, I would like to request that more output listings of assemblers and octal memory dumps, etc. be hex dumps and hex assembly listings. It will save space and time for all!!!

Hexidecimally yours,

Mark J. Nitzberg

4D 41 52 4B 20 4A 2E 20 4E 49 54 5A 42 52 47
115 101 122 113 040 112 056 040 116 111 124
132 102 105 122 107

15 South Dr.

East Brunswick, NJ 08816

P.S. Believe it or not, I am sixteen (0fh, 026q) years old by some coincidence. Also note that sixteen more characters were required to represent my name in ASCII octal than hex!!

DR. DOBB'S JOURNAL
COMPUTER
Calisthenics & Orthodontia
Running Light Without Overbyte

DATALIGHT C DELIVERS PERFORMANCE

DATALIGHT C \$60 ■ THE DEVELOPER'S KIT \$99

The **DATALIGHT C Compiler** is a performer through and through. From the UNIX System 5 language with the latest ANSI extensions (prototyping), to the top compile speed, with the understandable error interface, on to the tight/fast code — the Performer shines. Performance comes in two sizes: THE **DATALIGHT C Compiler**, and the **DEVELOPER'S KIT**.

DATALIGHT C provides you with a full range of features, like full 8087 and software floating point, a UNIX-style **MAKE** program, one-step compile/link, and UNIX-like tools in source form. You also get automatic .COM file generation, MS-DOS compatible object files, third-party library/debugger support, and much more.

The **DEVELOPER'S KIT** provides the extra features required by the serious programmer. The **DEVELOPER'S KIT** allows you to build programs as BIG as the memory in your PC. You can also tailor your applications to your special PC or ROM-based needs using the start-up and library source provided.

And what do our users think? They love us! In fact, the ones who

own higher-priced compilers love us the most because they know what **PERFORMANCE** really is!

"This is a sharp compiler! . . . what is impressive is that DATALIGHT not only stole the compile time show completely, but had the fastest Fibonacci executable time and had excellent object file sizes to boot!"

Chris Skelly

COMPUTER LANGUAGE

"I have and actively use Microsoft C version 3.0, Mark Williams C, and Lattice version 3.0, in addition to your compiler. Of all the compilers that I have used yours really stands out as the best package."

Matthew Brandt

TANGENT TECHNOLOGIES

"Of all the MS-DOS C compilers available, we have chosen DATALIGHT for all our development."

Keven Smith Drew Gileson

TRAVELING SOFTWARE

So why wait? You can have the Performer, a 30-day money-back guarantee, and the call is on us. So order now!

DATALIGHT C (Ver 2.10)

- Full UNIX System 5 C Compiler with ANSI extensions.
- Fast/tight code.
- 8087 & software floating point.
- Full UNIX compatible library.
- **MAKE** program with macros, dependency checking, and MS-DOS internal commands.
- **DLC** one-step compile/link program.
- Tools in source (diff, cat, pr, wc, rm).
- Powerful utilities in source form.
- Compatible with MS-DOS linker.

DEVELOPER'S KIT (Ver 2.10)

- All features of **DATALIGHT C**!
- Third-party library support.
- Multiple memory model support
 - Small 64k code 64k data
 - Data 64k code 1Meg data
 - Program 1Megcode 64k data
 - Large 1Megcode 1Meg data
- Complete **SOURCE CODE** for libraries.
- Complete **SOURCE CODE** for start-up routine.
- ROMable code.

MS-DOS is a trademark of Microsoft.
UNIX is a trademark of Bell Labs.
Lattice C is a trademark of Lattice Inc.

Datalight

P.O. BOX 82441
KENMORE, WA 98028
(206) 367-1803

ORDER NOW! 30-DAY MONEY-BACK GUARANTEE.

YES, I WANT THE PERFORMER!!!

By phone call 1-800-628-2828 — Ext. 571 (Orders only)
Technical Information (206) 367-1803

I want _____ copies of **DATALIGHT C** (\$60)

I want _____ copies of the **DEVELOPER'S KIT** (\$99)

Add \$5 for shipping in US/\$15 outside US. C.O.D. (add \$2.50)

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

CARD # _____

EXPIRATION DATE _____

P.O. # _____ (attach copy)

Circle no. 203 on reader service card.



LETTERS

**Carew's Flames**

Dear DDJ,

David Carew, author of the June Viewpoint, "What's Wrong with C," may be interested to note that my C compiler (Manx Aztec) for the IBM PC produces identical code for the two fragments he supplies:

```
b = ++i
```

or

```
i = i + 1
```

```
b = i
```

One of Mr. Carew's main points in attacking C is that the former produces "radically better code." Obviously it does not.

Mr. Carew is right in his contention that C code can be difficult to understand and to maintain (although this is hardly news). C code can also be elegant and quite portable. I have converted several programs written by others for use with my particular compiler (some from Unix and CP/M) and have had very little difficulty understanding or maintaining the code involved.

Perhaps the most confusing aspect of Mr. Carew's article is his comparison of the output of C compilers to that of "an average production-quality optimizing compiler." Is there a word missing from this phrase? Presumably these compilers are compiling something. Tradition suggests

that it would be source code in some language. Apparently Mr. Carew thinks that some other high-level language produces tighter and faster code than C does but is hesitant to name it. We can only guess his reason for this.

Given that you have an application to write, the application will perform the same tasks no matter what language it's written in. The speed and compactness of the code are therefore a function of the quality of the compiler, not of the language. In my own very-high-level language (BOB), you can issue simple statements such as *compile standard mailing list* or *compile standard Word-Star clone* and the compiler does the rest. The code pro-

duced rivals that of the very best assembly-language programmers or optimizing compilers. Unfortunately the compiler (also called Bob) is often struck by periods of existential ennui during which he is unable to compile anything but is still able to code in C and leave things up to another somewhat less intelligent but much faster compiler.

Dr. Bob

444 Maple Ln.

St. Paul, MN 55126

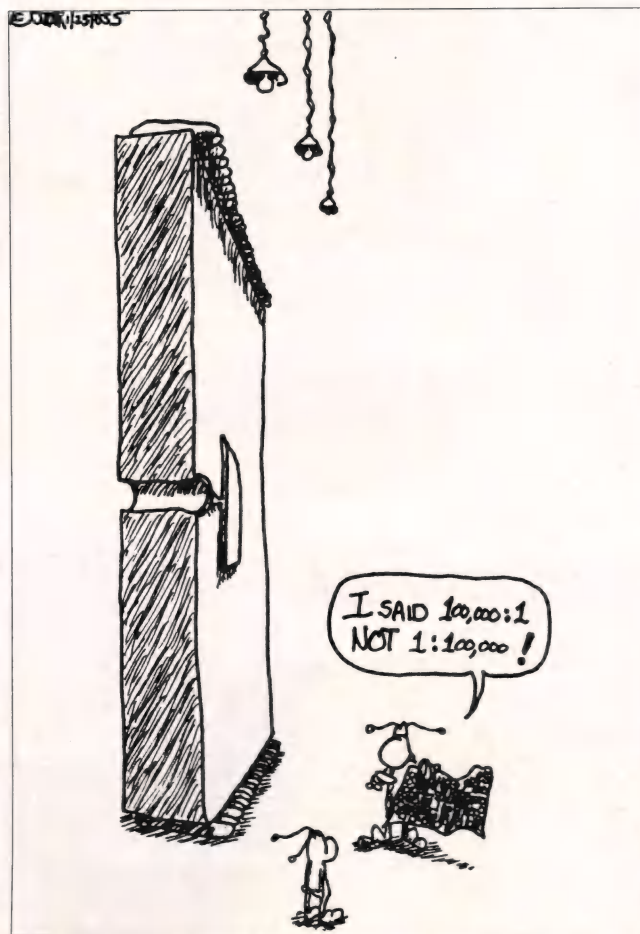
Dear DDJ,

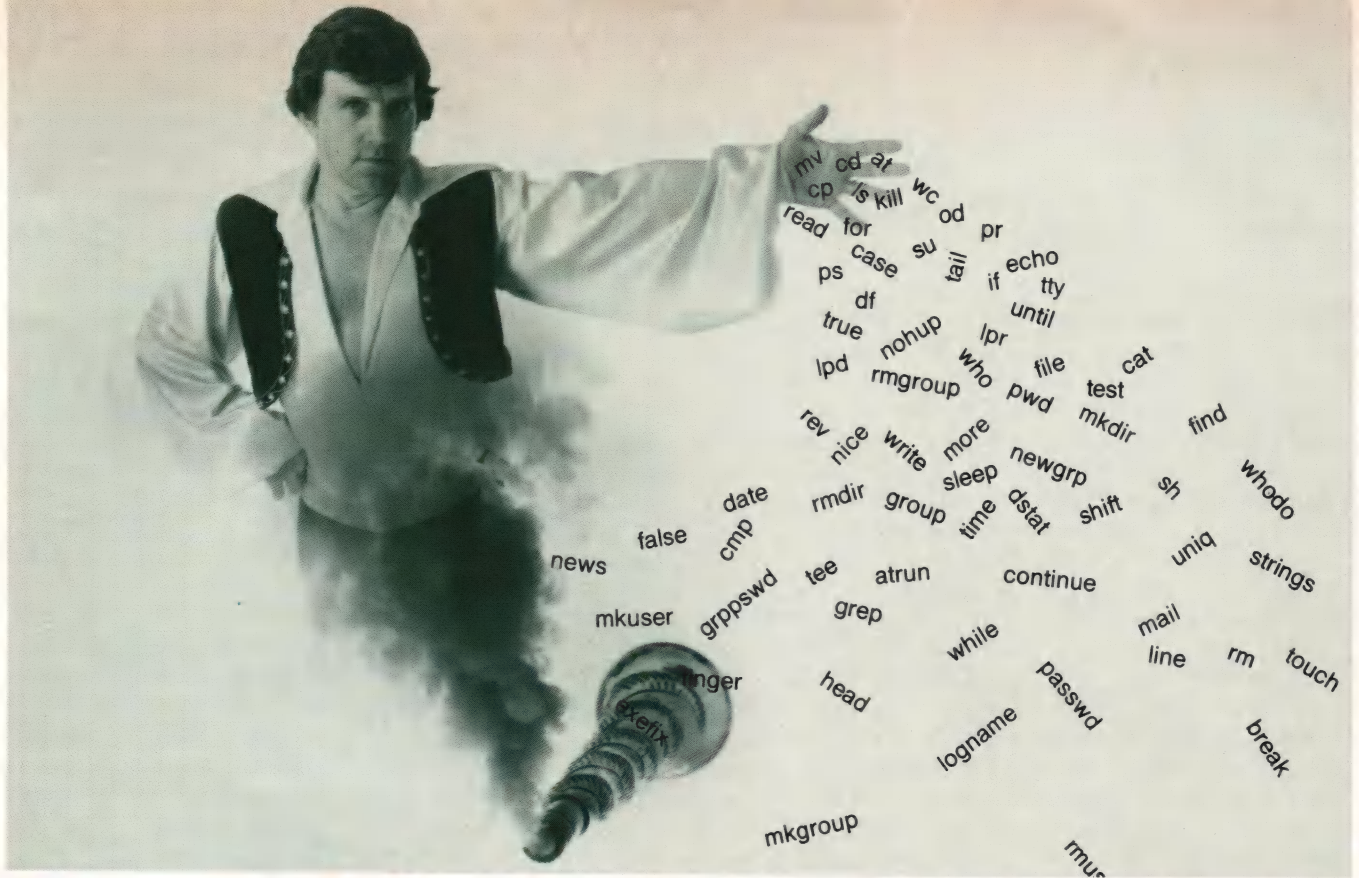
As a professional programmer—one of Bill Gates' crew, in fact—I feel called upon to respond to David Carew's article, "What's Wrong with C," in the June 1986 issue. Having used C

extensively in the past five years, I will readily admit that the language is not without its deficiencies. Mr. Carew, however, has not mentioned any. He confuses bad C compilers and bad programmers with flaws in C.

"C simply doesn't allow the use of standard compiler optimization techniques." This statement certainly comes as a surprise to the programmers who wrote the optimizer for the Microsoft C compiler. They were under the impression that detection of common subexpressions, constant folding, and peephole optimizations such as redundant jump elimination were standard optimization techniques. These are just some of the optimizations that the Microsoft C compiler does, and it is not the only optimizing C compiler available, either. It is true that certain constructs in C cannot be optimized safely, but these are the very constructs that produce efficient code when used properly. Further, there is nothing to prevent a programmer from writing C code using the "vanilla" constructs that are also found in Pascal, and there is nothing to prevent a good C compiler from performing sophisticated optimizations on such code.

C is "inefficient compared with the output of an average production-quality optimizing compiler." Mr. Carew makes this bold statement, but he offers no facts to back it up. I might as well say that I think people from Colorado tend to make more unfounded statements than do people from Washington. I present Mr. Carew as





Your wish is our commands.

UNIX operating systems have a well deserved reputation for their powerful command set. Now PCNX gives you the same powerful commands. And a lot more.

Built with Wendin's Operating System Toolbox, PCNX is the only multitasking, multiuser operating system that can put the popular features of UNIX on your personal computer (IBM PC, AT, XT, or true compatible) for under \$100.

If you're already familiar with the UNIX environment you know what that means. If you aren't, PCNX is your chance to discover what you've been missing at a price you can finally afford.

Designed specifically for systems programmers, PCNX has all the features that have made UNIX the preferred environment for systems development. Commands that let you perform complex operations with a few keystrokes. Plus syntax that lets you compile programs in the background, handle piping and I/O redirection, and build shell script files.

And that's not all.

PCNX links the popular Bourne shell to Wendin's unique kernel. As a result, you have access to more than eighty enhanced system services built into all our Personal Operating Systems.

Like our other Personal Operating Systems, PCNX also runs well-behaved MS-DOS programs, is fully compatible with the MS-DOS file system, and uses existing MS-DOS compilers, linkers and utilities.

In addition, it comes with complete source code that lets you see for yourself exactly how the system works.

So if you've ever wished you had the power and flexibility of UNIX on your PC, why not stop wishing and order PCNX today.

PCNX. From Wendin. Only \$99.

Ask about our other products for the IBM PC and true compatibles.

PCVMS™

Multitasking, multiuser version of DEC's powerful VAX/VMS operating system. Runs most MS-DOS programs.

XTC®

The ultimate programmer's editor. Multitasking macro language plus multiple linkable windows and buffers.

OPERATING SYSTEM TOOLBOX™

Complete software construction set that lets you build your own multitasking, multiuser operating systems.

All products priced at \$99 with source code included.

DEALER INQUIRIES WELCOME

Foreign orders inquire about shipping. Domestic orders add \$5.00/1st item, \$1.00 each additional item for shipping, handling, and insurance. We accept Visa/MC, American Express, COD, and Bank Drafts drawn on U.S. Banks. Washington residents add 7.8% sales tax.

MS is a trademark of Microsoft, PC-DOS is a trademark of IBM. UNIX is a trademark of AT&T. VAX/VMS is a registered trademark of Digital Equipment Corporation.

Wendin and XTC are registered trademarks of Wendin, Inc. PCNX, PCVMS, Operating System Toolbox, and Personal Operating System are trademarks of Wendin, Inc.

WENDIN®
BOX 266
CHENEY, WA 99004

© Copyright 1986 Wendin, Inc. The people who make quality software tools affordable.

ORDER HOTLINE
(509) 235-8088
(MON.-FRI., 8-5 PACIFIC TIME)



Circle no. 112 on reader service card.

LETTERS

(continued from page 10)

evidence. One example is hardly conclusive, but it is far better than nothing.

"C's operator set is too rich." You might as well say, "The English language is too rich." As with any language, natural or otherwise, sensible people will use only those constructs with which they are familiar. Shakespeare and Churchill used English far better than I do. What a loss if they had been forced to write at my level!

"I am often struck by the impression that a given C program is an elegant example of C and its operators but misses the point as a solution." Perhaps Mr. Carew has never read any C written by competent programmers. A poor choice of algorithms makes for an inelegant solution no matter what language is being used. A programmer who becomes "distracted from the task of contriving an optimal solution to the problem at hand" is an unprofessional programmer—a hacker.

"When one construct generates radically better code, it is natural for the programmer to expend effort optimizing his or her use of the programming notation." If true, this is a valid point. It is simply not true, however. I present Mr. Carew's examples and the corresponding code generated by the Microsoft C compiler (Version 3.00):

```
b = ++i;
    inc WORD PTR [bp-2];i
    mov ax,[bp-2]      ;i
    mov [bp-4],ax      ;b
i = i + 1;
    inc WORD PTR [bp-2];i
b = i
    mov ax,[bp-2]      ;i
    mov [bp-4],ax      ;b
```

As you can see, the code is identical.

"Better algorithms and data structures are far more important than is ideal use of a complex programming notation." Of course. What is Mr. Carew's point? Will reading Kernighan and Ritchie somehow destroy a programmer's judgment? The example Mr. Carew gives is meaningless. Any programmer who would choose a selection sort over quicksort when sorting more than a few dozen elements is not a good programmer. Does it matter what language is being used? Would Mr. Carew care to wager that a carefully coded quicksort in compiled BASIC will beat a carefully coded quicksort in C?

"The investment in learning C is so high." Again a statement made with no support. I can speak only from personal experience. C was the first structured language I learned. It took me a day or two to begin writing correct code. My major obstacle was poor diagnostic messages from the compiler. This is a compiler implementation issue, not a language issue. I do not think my experience was either unreasonable or atypical. Of course, learning to use any language well takes more than a couple of days.

Mr. Carew's complaints are misdirected. They apply to poor C compilers and poor programmers but not to the C language. Mr. Carew invites controversy by making statements without attempting to provide any substantiation. The gentleman is certainly entitled to his opinions, but by failing to support them, he sounds like a crank up on a soapbox.

The opinions expressed

herein are my own and do not necessarily reflect those of my employer.

Pete Stewart
Microsoft Corp.
16011 N.E. 36th Way
P.O. Box 97017
Redmond, WA
98073-9717

Dear DDJ,

I am concerned about your Viewpoint forum. As an educated DDJ reader, I expect copious facts or observations to support a position presented. D. Carew presents unsupported assertions. The "brutal fact" is that no DDJ quality examples of optimizing compilers vs. C were given. Second, he suggests that mediocrity is better than elegance or efficiency. Can he be serious? Would you adopt his view? I wouldn't.

Dr. Barr E. Bauer
9 Stone Ave.
Elmwood Park, NJ 07407

David Carew replies:

Dr. Bauer believes that I suggested mediocrity is better than elegance and efficiency. I did not mean to do so. I did mean to suggest that productivity is better than elegance and efficiency, with the proviso that in general efficiency is not sacrificed when C is given up and that elegance is much in the eye of the beholder.

It is perhaps lame to point the finger elsewhere in defending one's own viewpoint. In my original submission, however, I had at least one example cited and made mention of Modula-2, Edison, occam, and (I believe) Ada as alternatives available for microcomputers that may be more productive than C, or more efficient than C, or both. The copy I refer to was cut out of the final piece. Perhaps this was done because examples are

so obvious and plentiful. Almost everywhere you look, you can find examples of optimizing compilers with higher level syntax that equal or beat C in standard benchmarks.

In addition to those mentioned above, the VMS BASIC compiler beats portable C on the VAX. On virtually every operating system that has them to compare (except Unix!), hoary old FORTRAN and even COBOL can be found outbenchmarking C.

In fact, what you get when choosing C is portability and a certain low-level, "no-limits-on-what-I-can-do" feeling. (Perhaps this is what people mean when they rhapsodize about C's "power" and "elegance.") From 1975 to perhaps 1984 or 1985, this was indeed a rare combination. It is now not so rare. All choices are trade-offs. What you give up in choosing C's portability and "power/elegance" is:

1. Efficiency of the compiler's output object code.
2. Productivity considering the entire life cycle of the software (80 percent maintenance, remember!).

It is curious to me that everyone seems willing to concede point 2, which is much more important in terms of total dollars cost, while stongly denying that point 1 has any validity.

As for the embarrassing fact that my example C fragments produce identical code, I can only say that it proves the obvious: I am no C wizard. The basic point is that C is a notation that favors powerful complexity over optimizable simplicity. Those more familiar with C can surely fill in a good example for my bogus one. The expert's terse and idiomatic C does

HAUPPAUGE

Is Getting A Fast Reputation.



HAUPPAUGE started earning a fast reputation with their 87 Math Pak, the combination of an 87 chip and 87 Software Pak that's been accelerating PC math since 1982.

Next came their racy 287 FAST/5, a math coprocessor module with its own 5MHz clock, speeding up PC/AT math by 25%. (Pictured above.)

Now, Hauppauge Unveils the 287 FAST/10...

Our newest math coprocessor for the PC/AT, the 287 FAST/10 moves out at 10MHz—more than doubling the speed of each floating point math operation. The FAST/10 accelerates AutoCad, 1-2-3, Symphony, Turbo Pascal, Framework and more. The FAST/10 also runs in PC/AT compatibles including the Compaq Deskpro 286, Sperry PC/IT and most 286 accelerator boards.

...And the 87 Software Pak Version 6.0

Designed to steal the heart of programmers, the 87 Software Pak supports IBM's BASIC Compiler 1.0 and 2.0, and Microsoft's QuickBASIC, executing math-intensive programs up to 20 times faster! The 87 Software Pak also performs FFT's and Matrix operations. For example, a PC (or PC/XT) with an 87 Chip and 87 Software Pak can perform a 512-point complex FFT in just 1.1 seconds. What's more, a PC/AT with a FAST/10 inverts a 25 by 25 element matrix in under 1 second.

HAUPPAUGE Math Coprocessors

287 FAST/10 10MHz math coprocessor for PC/AT and compatibles...\$469

287 FAST/8 8MHz math coprocessor for PC/AT and compatibles...\$379

287 Chip PC/AT math coprocessor—runs at 4MHz in the old model IBM PC/AT and 5.33 MHz in the new model PC/AT...\$219

87 Chip Math coprocessor for IBM PC, PC/XT and compatibles...\$129

87 Chip Math coprocessor for 8MHz PC compatibles...\$195

HAUPPAUGE Math Coprocessor Paks

87 Math Pak V.6.0 87 chip and math coprocessor software support for IBM BASIC Compiler 1.0, 2.0 and Microsoft's QuickBASIC. Plus, Matrix and FFT support, one year of free updates, complete source code and "8087 Applications and Programming".....\$279

87 Software Pak V.6.0 Math coprocessor software support as in the 87 Math Pak, but without 87 chip.....\$180

87 QB PAK Math coprocessor support for Microsoft QuickBASIC and IBM BASIC Compiler.....\$ 69

Recalc+ Math coprocessor support for 1-2-3 version 1A.....\$ 95

HFT+ Complete Hayes Fourier Transform Package...\$125

The 287 FAST/10 Doubles Your PC/AT's Math Speed!

Help your PC/AT get a fast reputation with Hauppauge's new 287 FAST/10. Call today, or contact your local computer dealer to learn more about Hauppauge's racy product line. And ask for "87 Q & A," our free booklet on math coprocessors.

Hauppauge Computer Works, Inc.

358 Veterans Memorial Highway, Suite MS1,
Commack, New York, USA 11725

516-360-3827 • TELEX: 262939-HCW

Hauppauge ("Haup-pa")

We acknowledge the following registered trademarks: 1-2-3 and Symphony: Lotus Development Corporation; AutoCad, AutoDesk, Inc.; IBM: PC, PC/AT, PC/XT; Compaq: Deskpro; Sperry: PC/IT; Texas Instruments: Business Pro; Ashton-Tate: Framework. Circle no. 274 on reader service card.

LETTERS

(continued from page 12)

produce better results than the beginner's C, coded as though it were Pascal. Expert and beginner alike may well have a tendency to spend time exploring and exploiting the complex notation and the preprocessor at the expense of mastering the application problem.

Who is DDJ for?

Dear DDJ,
I'm not a serials cataloger, but I do sympathize with Dave Sullivan's comments in the July issue's Letters column. Presumably, frequent variations in the title of a journal are a valuable marketing tool, but they are also a librarian's nightmare. It is all the more aggravating because, as far as

I can tell, the editorial content of the journal has not changed as much as the title has.

Putting all that aside, I wouldn't miss a single issue. Even though I am not a professional programmer, more of what I am interested in, and need, is in *DDJ* than in any other source. Whatever it is you're doing, please keep it up. Just cool it on the title changes, OK?

Bruce B. Cox
Automation Committee
Linda Hall Library
5109 Cherry St.
Kansas City, MO 64110

OS-9 Bugs

Dear DDJ,
I must respond to the letter written by Tim Harris of Microware Systems Corp. that appeared in May's *DDJ*. I took a chance on a medium-size project that

involved porting a DBMS to a multiuser environment under OS-9. Although I agree that the design intention of OS-9 is decent, the implementation is poor, laden with bugs, and backed with poor customer service. I have complaints about both OS-9's operating system and its implementation of C.

The OS-9 disk formatter utility, for example, has a bug that prevents it from being used with more than one sector per allocation cluster. This is not documented, Microware has no intention of repairing this bug before the next release, and it cost me several days of my time. OS-9 has a flawed file system, and the disk-access method used by OS-9 is also extremely slow. I benchmark it at from 1 to 100 times slower than equivalent Unix machines. The idea of building a disk-intensive application for a client on OS-9 makes me shudder.

Now for the real problems. The C compiler and its associated library are full of bugs. Last Friday I lost half a day because the library function *tsleep()*, which supposedly provides timed delays, is highly nonlinear in its function and, at some point determined by the system clock speed, suffers from a discontinuity that severely shortens the response time. I was using this function to determine whether a multicharacter key had been pressed on the keyboard and was attempting to wait a hundredth of a second—instead the wait was approximately 1/3,000 of a second. This behavior is not documented.

Earlier last week I ran into a problem with the ***= operator used with a variable of type *long* that completely hung the system. I

lost half a day tracking that down. The week before that, I ran into a bug in which floating-point cast to integers fail to trigger an *if()* expression correctly. Some problem crops up once or twice a week with this compiler or its library.

The current compiler release came out in February. The previous compiler was worse. If you used parentheses in a certain way, it would get lost and not even perform integer division correctly.

My biggest complaint is that Microware will not remedy what is broken. If I get stuck, it will not fix the broken compiler and ship me an update. It claims it is too big to be bothered with sending out updates between major releases. My claim is that its software is too broken not to.

Beware of OS-9 if you value your business and your sanity.

Heitzso
MetaMedia Inc.
P.O. Box 292
Atlanta, GA 30301

Correction

Listing Seven of the August 1986 C Chest contained a bug that could cause the insert function to fail on an attempt to insert a conflicting node. To fix the problem, add the line *h = 0*; immediately below line 68 (page 92).

DDJ

Publication Quality Scientific Graphics

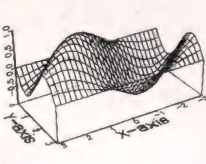
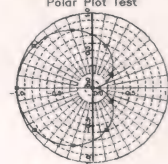
Graphic 3.0

Over 100 C routines make scientific plotting easy

- linear, log, & polar plots
- bar charts & Smith charts
- contour plots with labels
- 3-D curves, 3-D surfaces
- 4 curve types, 8 markers, errorbars
- 14 fonts, font editor
- unlimited levels of superscripts
- 4096 x 3120 resolution in 16 colors on EGA, Tecmar, Sigma boards
- zoom, pan, window and merge plots
- high resolution printer dumps

SOURCE INCLUDED for personal use only
\$350. Demo \$8

256k, IBM, AT&T, Corona PCs, DOS 2.xx, 3.xx
Most boards, printers, and plotters supported
Microsoft, Lattice, DeSmet, Aztec, C86 compilers


Scientific Endeavors Corporation

Route 4, Box 79 Kingston, TN 37763 (615) 376-4146

Circle no. 210 on reader service card.



USE THE BRAINS YOUR IBM WASN'T BORN WITH.

Right at your fingertips in CompuServe's IBM® Forums.

In the **IBM New Users Forum** you'll swap ideas with other new PC users, learn to use Forum features, and pose even basic questions to PC experts.

Our **IBM Junior Forum** gives PCjr® users a reliable source for tips on software, hardware, telecommunications, games and other interests.

In the **IBM Software Forum** you'll trade tips with other IBM PC and AT users on utility software, word processing, DOS and other operating systems.

Visit the **IBM Communications Forum** for advice on the features and compatibility of communications software and hardware, PC Bulletin Boards, micro-mainframe interfaces and more.

The **IBM Hardware Forum** addresses hardware topics of all types, plus product updates and announcements.

Easy access to free software, including FREE uploads.

- Download first-rate, non-commercial user-supported software and utility programs.
- Upload your own programs free of connect time charges.
- Take advantage of CompuServe's inexpensive weeknight and weekend rates (when forums are most active, and standard online charges are just 10¢ per minute).
- Go online in most major metropolitan areas with a local phone call.
- And receive a **\$25.00 Introductory Usage Credit** with purchase of your CompuServe Subscription Kit.

Information you simply can't find anywhere else.

Use the **Forum Message Board** to send and receive electronic messages, and pose specific questions to other IBM and compatible owners.

Join ongoing, real-time discussions in a **Forum Conference**.

Search our unparalleled **Forum Data Libraries** for free software, user tips, transcripts of online conferences and more.

Enjoy other useful services like:

- **Popular Computer Magazines**—electronic editions, for your reading pleasure. Including *Dr. Dobb's Journal* and *Computer Language*.
- **Other CompuServe Forums**—supporting LOTUS® products like *Symphony™* and *1-2-3.™* Borland International®, Ashton-Tate®, Digital Research®, MicroPro®, Microsoft®, Software Publishing® and others.

All you need is your IBM or IBM-compatible computer and a modem ... or almost any other computer.

To buy your Subscription Kit, see your nearest computer dealer. Suggested retail price is \$39.95. To receive our free brochure, or to order direct, call 800-848-8199 (in Ohio, call 614-457-0802). If you're already a CompuServe subscriber, type GO IBMNET (the IBM Users Network) at any ! prompt to see what you've been missing.

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220
800-848-8199
In Ohio, Call 614-457-0802
An H&R Block Company

DDJ ON LINE

We've opened a new section on the SIG called *TEACHING/LEARNING*. The message board of this section is a place where expert programmers can give advice to absolute beginners. (Where would you begin? Which language? What books?) It's also for experienced programmers who wish to learn a new language.

The Data Library will primarily contain threads from the board and reference lists.

—Levi Thomas (*SYSOP)

C Chest

The following discussion took place on our C Chest message board on the DDJ Forum:

#: 3092 S1/C Chest

Fm: Bill

To: all

I'm at that stage of experience with C where I learn about the language mostly by analyzing my mistakes. Here's one I need some help with. Yesterday I encountered what I thought was a compiler bug in Aztec C86. I was getting "symbol re-definition errors" for the second declaration of *errorvar* in a piece of code typified by the following fragment:

```
int fn1 ( )
{
    int errorvar ;
    int dummyvar ;
    fn2(dummyvar) ;
}

void fn2 (errorvar)
int errorvar ;
{
}
```

Manx's explanation: "That's not a bug; it's a feature" (of the language). The (legitimate) error message

was actually produced by my redefinition of *fn2*; that Aztec C86 didn't detect and flag it until the following line was "interesting." I'd be inclined to agree with that; in fact, after several hours of tearing my hair and several more trying to reach Manx by phone, I might have used a slightly stronger adjective. To continue with the explanation: "Because *fn2* wasn't formally declared before being invoked, it defaults to *int*. The later declaration as *void* triggers the 're-definition error' diagnostic." I should (Manx said) either move *fn2* above *fn1* in the source or else declare *fn2* as *void* within *fn1* before invoking it.

Frankly, this strikes me as nonsense. If I craved a language that would kick me around in this way, I'd use Pascal. Bottom line: Will/should this code be flagged as incorrect by any standard compiler? If so, and if there's a good reason for including this "feature" in the standard, would somebody explain it?

Fm: Chris [IBMNET]

To: Bill

Sorry to say, Manx is right. This is one of the very few cases where C compilers do any type checking at all. Any reference to the result of an as-yet-undefined function is assumed to be an *int*, and you'll get some kind of diagnostic if that later turns out to be wrong. The company's two suggested resolutions are also correct. Not sure why the compiler waits until the *errorvar* declaration. . . .

Fm: Bill

To: Chris

Maybe I expressed my point badly; I realize there

are times the compiler must make assumptions, if only to clear the procedure stack. Where externals are concerned, I have no problem with default typing. But in my example, the compiler does/could know the function type because it's explicitly declared *void* a few lines later on. Doesn't it strike you as odd to talk about assigning a "default" type to a symbol that's explicitly typed in the same source module? As my error shows, a language that works in this way will frequently be unable to handle forward references in an intelligent way (that is, without requiring contortions on the part of the programmer). I have apparently escaped being bitten by this up to now through sheer luck; it gives me a spooky feeling to think that I could go back and edit working code, changing nothing but the order in which functions appear in the source module, and produce numerous compile-time errors.

Fm: Larry

To: Bill

A couple of things—the reason why standard C doesn't catch this is that it doesn't do forward references. That is, *all* symbols, whether functions or variables, must be predefined to non-*int* if they are to be used as such. The reason I say standard C is that it is not infeasible that you might write a multipass C compiler; but the Kernighan/Ritchie, Harbison/Steele, and ANSI C standards/references all describe C in this manner.

Second, the way that I try to code is that all variables, whether functions or not, get declared explic-

itly—I am not one for depending on a compiler to do things such as declarations itself. Even things such as *strcpy()* generally get placed in a preamble to my code. If I have macros that call subroutines (declared within a header), that is where the function is declared. I am beginning to write:

```
#ifdef lint
:
:
#endif
```

clauses as well, putting info in so that lint knows what is going on for personal functions.

Note that in Pascal you have only the choice of declaring a function before using it (in general)—there is no mechanism I am aware of to allow simple declarations of a later occurring function. At least in C, you can briefly tell the compiler how to handle the calls to the subroutine and then later actually go ahead and specify the subroutine.

Fm: Chris

To: Larry

Oops! You can make any Pascal function/procedure declaration *FORWARD*, which allows you to use as-yet-undefined blocks. The compiler just needs the parameter list and result type if it's a function. . . .

Fm: Larry

To: Chris

OK—I had never heard of it. Of course, I have only taken a few intro courses on Pascal and used it only on an Apple II (Apple's Pascal). . . . I shouldn't have spoken of that which I did not know for sure—sorry all!

Are you getting the most out of *Turbo Pascal?*

Do you believe Turbo Pascal is a serious programming language and not just a toy? Would you like to improve your skills and master the power and ease of Turbo Pascal programming? Then M&T has a publication for you: *Turbo Tech Report*.

This bimonthly newsletter/disk publication is written in the *Dr. Dobb's Journal* tradition, delivering high-quality technical reviews, in-depth articles, tips and tricks, and Turbo Pascal utilities, libraries, and source code on disk. Editor Namir Clement Shammas is dedicated to bringing you the best in Turbo programming, insight, commentary and gossip.

Each valuable issue will include:

- Reviews of Turbo Pascal-related products. You'll read about the latest Turbo Pascal third party software

developments, like Blaise Computing's new Turbo Power Tools.™

- In-depth articles on expanding your Turbo Pascal programming skills—on topics like 3D graphics, screen utilities, and memory resident programs.

- Applications

developed by other Turbo users and public domain Turbo Pascal software programs on disk.

If you're an expert Turbo Pascal programmer or a novice interested in expanding your Turbo skills, you need *Turbo Tech Report*.

Subscribe today at the special introductory price of just \$99—that's 33% off the regular price of \$150.

To order by credit card, call toll free 1-800-528-6050 ext. 4001 and ask for item 300.

Or mail the attached coupon with your payment to *Turbo Tech Report*, 501 Galveston Drive, Redwood City, CA 94063.

Introducing *Turbo Tech Report*, the newsletter for Turbo Pascal programmers. 6 disks with 6 newsletters for \$99 a year!

Turbo Pascal is a trademark of Borland International Inc.
Turbo Power Tools is a trademark of Blaise Computing Inc.

Circle no. 119 on reader service card.

Fm: Bill

To: Larry

Actually, K & R is considerably stricter than that. (After finding nothing under *definition*, *declaration*, *type*, *scope*, and so on, I'd about decided this wasn't addressed explicitly. (I finally stumbled across it under *functions returning a non-int*. Unless declared externally to all functions, a noninteger function has to be declared within each and every other function that references it.) Although your practice has obvious advantages for debug and documentation, it still seems strange to me to be forced to declare a function more than once. Do I understand correctly from your comments that C

compilers are always or almost always single-pass jobs? Being even more ignorant about implementations than about the language itself, I'd assumed the opposite—one reason why this behavior seemed so bewildering.

Fm: Larry

To: Bill

From my experience, in Unix at least, the C compiler does the following: invokes a preprocessor (which makes a single pass through the file), invokes a program that converts the C languages into a tokenized form, invokes a token → assembler program, assembles the program, optimizes the program (this step may actually take place before the assembly—I forget at this moment), then links the pro-

gram. Each step, except perhaps the final two, seems to be a single pass through the current file. Note that in many ways, the link step itself is a single pass—that is, the ordering of object modules/libraries within Unix is usually very critical. It may make for a somewhat faster link, but I would prefer a multipass approach here as well as in some other places in the process.

Fm: Shira

To: Bill

Bill—It's worse than that. You can get run-time errors if, for example, the function returns a *long* and you don't predeclare it. And (forgive me if I'm being pedagogic here), assigning the result to a *long* (as in *longvar = lfun ();*) makes it look OK but doesn't im-

prove the result. As I understand it, the new ANSI standard is moving toward stronger type checking, so this feature is probably permanent.

It's not so bad, really . . . encourages good programming practices for one thing (what if you decide later to move *fn2* to a library?).

Does anybody know, btw, how C++ handles this? And is there a PC implementation of C++ yet?

Fm: Lenox

To: Shira

In ANSI C (which, of course, is currently only a "proposed" standard), a program *must* declare any function before it is actually defined to be considered a "strictly conforming" program. Here is an example:

FOR C
AND UNIX

WINDOWS FOR DATA™

MS WINDOWS
COMPATIBILITY

DATA ENTRY WINDOWS MENUS HELP

Windows for Data does the hard jobs that others can't — we **guarantee** it. Makes standard display and entry tasks easy. Reliable. Compact. Portable.

DATA ENTRY: The most complete and flexible data entry system on the market. Pop-up data-entry windows; field types for all C data types, plus decimals, dates, and times; auto conversion to and from strings for all field types; system and user-supplied validation functions; range-checking; scrollable context-sensitive help; required and must-fill fields; programmer-definable edit keys, field types, and field masks. Read field by field or auto-read all fields. Branch and nest window forms. **Virtually every capability of WFD can be modified to meet special needs.**

WINDOWS: WFD is built upon and includes **Windows for C**, the windowing system rated #1 in PC Tech Journal (William Hunt, July 1985). WFC now has more features than ever, including automatic full compatibility with Microsoft Windows and TopView.

UNPRECEDENTED FLEXIBILITY



As many possibilities as Vermont in June.

MENUS: Build multi-level menus in the format of Lotus 1-2-3, Macintosh, or a style of your own choosing.

HELP: Build context-sensitive or menu-driven help systems. Display text in pop-up, scrollable windows.

UNIX, DOS, OR BOTH

WFC and WFD provide source code compatibility between PCDOS and UNIX.

OUR CHALLENGE AND GUARANTEE

If you have an application where no other tool can do the job, try **Windows for Data**. If it doesn't help you solve your problem, RETURN FOR A FULL REFUND. YOU MUST BE SATISFIED.

WINDOWS FOR DATA WINDOWS FOR C

PC DOS*	\$295	\$195
XENIX-286	\$595	\$395
UNIX	CALL	CALL

Call for **FREE Demo diskette**.

*All popular C compilers; no royalties.



**Vermont
Creative
Software**

21 Elm Ave.
Richford, VT 05476
**802-848-7738,
ext. 31.**

MasterCard & Visa Accepted. Shipping \$3.50
VT residents add 4% tax.

Btrieve.[®]

The Programmer's Choice.

When you're serious about application development, there's just one choice for file management: Btrieve. With what *Computer Language* calls "near mainframe functionality¹", Btrieve sets the file management standard for PC applications. With Btrieve loaded in your PC, your programs can use simple subroutine calls to retrieve, store and update records.

B-tree based for high performance. Performance is all-important, especially as your database grows. That's why Btrieve implements the b-tree file structure—the fastest, most efficient method of accessing data.



Interfaces to C, BASIC, Pascal, COBOL. Don't waste time programming in awkward fourth generation languages! With Btrieve, simply use the languages you know best—and write applications the right way. Over 15 language interfaces available.



Multi-user versions for LANs and Xenix. When your applications need to network, count on Btrieve. A single version runs on all DOS 3 LANs, including IBM PC Network and Novell Advanced Netware. Btrieve is also available for Xenix and multitasking operating systems such as MultiLink Advanced, Microsoft Windows and IBM Topview.

Built-in security features. Lock up sensitive data with Btrieve's password protection and unique data encryption scheme—especially useful in local area networks.

Thorough documentation, easy implementation. Getting started with Btrieve is easy: the manual is packed with examples of every Btrieve function in BASIC, Pascal, COBOL and C.

Database queries, report writing. Add Xtrieve™ to your Btrieve applications for a fully-relational DBMS. Xtrieve's menu-driven interface gives your users the on-line query capabilities they need—without programming. Add our report writer option to produce custom reports and forms.

No royalties.
Need we say more?



Fault tolerant. Btrieve insures against database disasters. Two levels of fault tolerance guarantee data integrity during accidents or power failures—even if lightning strikes. No extra programming required.




SoftCraft

P.O. Box 9802 #917 Austin, Texas 78766 (512) 346-8380 Telex 358 200

Suggested retail prices: Btrieve, \$245; multi-user Btrieve, \$595; Xtrieve, \$245; multi-user Xtrieve, \$595 (for report generation, add \$145 for single-user and \$345 for multi-user). Available from SoftCraft and selected distributors. Requires PC-DOS or MS-DOS 2.X, 3.X, Xenix. Btrieve is a registered trademark and Xtrieve is a trademark of SoftCraft Inc. ¹From Computer Language, November 1985.

```
extern char *foo(void);  
/* This is the declaration */  
static char nonsense;
```

```
char *foo()  
/* This is the definition */  
{  
    return (&nonsense);  
}
```

In practice, I believe that the intention of this rule is to get you to declare everything in a header file for every module that uses *foo()* to *#include*. Then the compiler can tell if there are any type mismatches in either the use or definition of a given procedure.

I believe that C++ requires everything to be declared before it is actually defined. And I haven't heard of any PC implementations of C++: The compiler is just too much of a memory hog right now.

Fm: Jeff
To: Bill

As the others have pointed out, C requires variables and functions to be declared before they are used. Pascal usually does too—and for the same reason—so you can do the compilation with a single-pass parser. Assuming *int* for nondeclared variables makes it possible to pass off forward references to the assembler and/or linker.

C compilers are not really single-pass programs; they usually involve three to five passes not including the linker. Pass 1 is the pre-processor and pass 2 the parser. Pass 3 and beyond are where compilers start to vary: Some optimize the output from the parser before feeding it to the code generator; some optimize after the code is generated. Others do both, and many

just ignore optimization completely.

Next comes the assembler pass and then the linker. Note that none of the above has to be a separate program; some compilers have all the passes built into a single program and just call each pass in turn on the current code line.

The important thing to remember is that the compiler knows nothing about what happens after the current line being compiled. It only knows what it is doing now and what is in the symbol table. If a function has not been declared explicitly, it is not in the symbol table yet. Because the compiler must know what it returns to continue, it defaults to *int* and makes a symbol table entry that says so.

When it got to your declaration of *fn2()*, all it had was that symbol table entry to work with, and so it blew up.

Fm: Sam
To: Bill

After spending 15 minutes reading this thread, I'm inclined to comment. I like C, but I have to agree that explicit type declaration for functions is inconvenient. Not only that, it is a potential horror show for portability. Two cases in point (which happened to yours truly in his C programming infancy): Wrote a program using *atol()* on a Z8000-based machine. Worked like a charm. Ported it to a VAX. No go. Problem? I never declared *long atol()*; at the top of the program—making *atol()* assumed as *int*. On the Z8000, the byte order let me get away with it. Neither compiler complained a bit. Second case is the new type *void*. I have written many programs compatible with Unix, Version 7. Now, under System

V, I have a problem. I never had to declare *exit()* before! System V says *void exit()*; and Version 7 says *int exit()*. Why do I care? Because after my first experience, I made a law for writing portable code: *run lint and fix every error!* Now, of course, dear lint complains about *every* *exit* call under System V. And speaking of lint, does the ANSI spec solve the *malloc()* problem? *Malloc()* is defined to return a pointer for any valid data type, but lint insists that you can't rightfully typecast a *(char *)* to a *(struct *)*. I think C needs a type to complement *void*—*valid*. *Valid *ptr*; would mean "ptr can point to any data element." Whew. Said more than I thought I would. Good thread.

Fm: Larry
To: Sam

The ANSI standard proposed the type *void ** with the meaning that it is a pointer of generic type, sized and aligned to match any other type. Don't ask me how it plans to pull that one off for machines with different-size pointers. . . .

Fm: Allen
To: Bill

I realize I'm replying a little late in the thread, but there are several things that no one's mentioned yet. First, if the compiler processed forward references in the way you suggest, it would either have to go through the input twice (like an assembler does it) or keep elaborate tables around for resolving these references. Either way the compiler would be slower. C compilers are indeed single pass, and the language is designed in the way it is to make this possible.

As for the duplicate declarations, don't confuse a

declaration with a *definition*. A *declaration* is an announcement. All it does is announce the existence of an object to the compiler. It's something like a pseudo-op. That is, a *declaration* gives the compiler information that it will use to update its symbol table. A *definition* on the other hand actually allocates space for an object and generates a label associated with that object. The choice of words is rather unfortunate here. I'd prefer something such as *definition* and *allocation*. The usage stems from K & R, and no one's seen fit to challenge them.

You'll notice that Pascal doesn't allow any forward references at all. Subroutines have to be declared before you can use them. C isn't really a high-level language; it's a very fancy assembly language and should be treated as such.

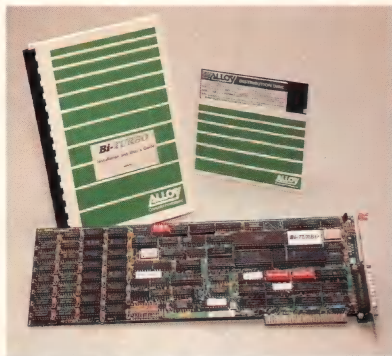
Fm: Shira
To: Allen

Allen—Thanks for the very clear distinction between declarations and definitions. Those of us who deal with multiple languages always seem to have trouble with these terms, but I've printed your message out and I won't mix up these terms again! Btw, by *these terms* I mean technical terms used differently by the developers of different languages. In, say, PL/I, a declaration normally allocates storage.

DDJ

Vote for your favorite
feature/article.
Circle Reader Service No. 1.

When You Need To Do Two Things At Once — ***Bi-TURBO***™... from ALLOY



Alloy's Bi-TURBO combines the features of today's best-selling boards and adds distinctive Alloy engineering experience to create the first accelerated dual-tasking card for personal computers. Bi-TURBO is the *complete* dual-tasking solution, with everything you need to be twice as productive.

- Adds power to PCs, XT's and even AT's
- Adds a full 640 KB of "second task" RAM, with its own 8 MHz processor
- Adds 256 KB of RAM-based disk cache to speed disk access for both tasks
- Easy-to-use dual tasking software — switch *full screen* windows with a keystroke!
- Print spooler, with *multi-printer* capability



With One PC and Bi-TURBO, Do the Work of Two PCs

You're in the middle of a three-hour mail merge and you need to run a spreadsheet — NOW! Bi-TURBO to the rescue! Press one key on your keyboard and you're running the spreadsheet — while the mail merge task continues to run *at full speed*.

You get both jobs done, and Bi-TURBO's RAM-based disk caching actually **SPEEDS UP** the mail merge!

\$995! — Bi-TURBO Saves Money and Time.

Bi-TURBO gives you the features and benefits of multi-function cards

and windowing software — *and much more performance* — for less.

Bi-TURBO is the *complete* power upgrade for your XT or AT, with added 8 MHz processing power, added RAM, disk caching, print spooling, truly *easy-to-use* dual-tasking software and optional 8 MHz 8087 math co-processor.

Call your local distributor today . . . and Bi-TURBO.

Crystal Computers, Inc.
Lenexa, KS 66214
(913) 541-1711
Irving, TX 75063
(214) 929-1300

FA Components
Elmhurst, NY 11373
(718) 507-1444
Ft. Wayne, IN 46808
(219) 432-8540
Greenville, SC 29607
(803) 288-2422

First Source Distributing
Salt Lake City, UT 84119
(801) 973-0074

Micro Computer Distributors, Inc.
Huntington Beach, CA 92649
(714) 895-5841

Micro Wholesalers, Inc.
Hunt Valley, MD 21030
(301) 666-5300

PC Distributing, Inc.
Northbrook, IL 60062
(312) 298-1400

PGI Corporation
Tempe, AZ 85281
(602) 967-1421

Super Source
Norcross, GA 30071
(404) 441-3451

Vitek
San Marcos, CA 92069
(619) 744-8305
San Jose, CA 95131
(408) 436-8026

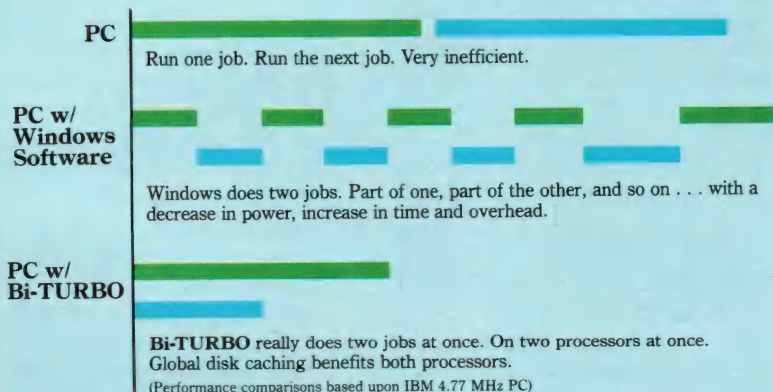
Vitronix Corporation
Westboro, MA 01581
(617) 366-1144

W4 Micro Distributors
Birmingham, AL 35209
(205) 945-8310

Western Micro Systems
Scottsdale, AZ 85260
(602) 948-4240
Mountain View, CA 94043
(415) 964-2050
Northridge, CA 91324
(818) 700-9922
Redmond, WA 98052
(206) 881-6737

Windows is a Registered Trademark of Microsoft Corp.

Bi-TURBO Doesn't Share Time, It Makes Time



ALLOY
Computer Products, Inc.

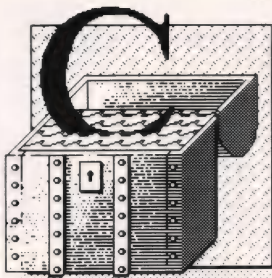
Alloy Computer Products, Inc., 100 Pennsylvania Avenue, Framingham, Massachusetts 01701. (617) 875-6100, TWX: 710-346-0394

Alloy Computer Products, Inc., 9 Executive Circle, Suite 240, Irvine, California 92714. (714) 261-7661

In Europe: Alloy Computer Products (Europe) Ltd., Cirencester, Gloucestershire, England. Tel. 0285-69571, Tlx: 43340

In Australia: Alloy Australia Pty. Ltd., Suite 3, 318-322 Stephenson's Road, Mount Waverley, Victoria, Australia 3149. Tel. 011-613-277-1333

More, a File-Browsing Utility



Microsoft, for some reason unknown to myself, used the name of the Unix utility *more* for the MS-DOS file-paging command. The Microsoft *more* is actually a subset of a Unix utility called *p* (for page); it's a subset because *p* accepts a list of files on the command line but Microsoft's *more* does not. The real *more* is a much more powerful file-browsing utility. It's useful anytime you want to look at a file but don't want to bother with an editor.

The program presented here is not an exact look-alike for the Unix utility. It does, however, support all the features of the real *more* that I use regularly. It also includes several commands not supported by the Unix version. Most important, it can go backward in the input file (even if the input file is *stdin*, provided that *stdin* is a redirected file or the end of a pipe). It can also move around in huge files. In fact, I wrote the program for this reason. I wanted to review the "nroffed" output of an entire book, a file that was a little more than a megabyte in size. The original for the book was split into about 20 smaller files, but the word-processed output was in a single file that my poor editor just couldn't handle. In addition to the ability to go backward, I needed several capabilities of the Unix *more*. In particular, I wanted to be able to execute my editor from within *more* and be back where I had left off when I

by Allen Holub

had finished editing. I've also added (at the suggestion of reader Fred Smith) the ability to search for a regular expression.

The command-line syntax is:

```
more [+<num>][file...]
```

The optional *+<num>* will cause

more to seek to character *<num>* before it starts printing. It's useful if you want to quit looking at a long file but then go back to it later. Note that this is different from the Unix version, which goes to line *<num>* rather than character *<num>*. Seeking to a character is much faster (because you can do it with an *fseek()* and don't have to process the skipped lines). The remainder of the command line consists of a list of files to process. If no files are present, *stdin* is used so you can use *more* at the end of a pipe (as in *ls | more* or *nroff -ms file | more*).

When the program starts up, it prints a page from the current input file and then waits for one of several commands. All these commands can be preceded by a count that will cause the command to be executed the specified number of times. A count doesn't always make sense (you wouldn't want to print the help screen *N* times), but it's supported for all commands anyway. The commands supported by my *more* are summarized in Table 1, page 24. They are:

- b—Go backward one page in the input (*Nb* will go back *N* pages). *More* prints a line on the screen and then prints the previous page. This way the program can be used on terminals that don't support backward scrolling.
- e—Go to the end of the current file.
- n—Go to the next file that was listed on the command line.
- o—Print the offset, in characters, from the beginning of the file. The offsets of both the top and bottom lines of the current screen

are printed. This command is useful in conjunction with the *+* command-line option.

q—Quit (return to DOS).

s—Skip one line. The skipped line is not printed. This command is usually used with a preceding count—for example, *100s* skips over the next 100 lines without printing them. You can still back up with the *b* command if you decide you really wanted to see the skipped lines after all. The current position in the file (represented as a percentage) is printed as a "mileage" indicator as lines are skipped.

r—For rewind. Goes back to beginning of the current file and prints the first page.

!—Waits for you to type a normal DOS command and then executes that command. Unix creates a shell to execute the command. I decided not to create a shell because of the additional memory required, so you can't execute a batch file or an internal command directly. You can do it indirectly by creating a shell explicitly, however. Use *command /c batchfile ...* for COMMAND.COM and *sh -c batchfile ...* for the shell. If you enter a carriage return instead of a command, the command used in the previous *!* is used this time. When the program terminates, *more* reprints the current page and then carries on as usual.

/—Prompts for a grep-like regular expression and then searches for a string matching that expression. The search starts at the first character after the end of the current screenful. The search terminates either when the string is found or when any key is pressed. As with the *!* command, the previous expression is used if a carriage return is entered at the prompt.

ESC—Input starts scrolling, without

First "language specific" DBMS written exclusively for C applications is also royalty free

"db_VISTA™ lets you easily build complex databases with many interconnected record types. . ."
Dave Schmitt, President, Lattice, Inc.

Designed exclusively for C, db_VISTA™ is a language specific database management system. Both single and multi-user versions let you take full advantage of C, through ease of use, transportability and efficiency.

Every Line of Code Written in C for C Programmers

All functions use C conventions so you will find db_VISTA easy to learn. db_VISTA operates on most popular computers, and because it is written in C it can easily be ported to most computers.

Royalty-Free, You Only Pay Once

Whether you're developing applications for a few customers, or for thousands, the price of db_VISTA is the same. If you are currently paying royalties for a competitor's database, consider switching to db_VISTA and say goodbye to royalties. To help you make the change over to db_VISTA, ASCII file transfer utilities are included. dBASE file transfer utilities are available as an option.

More from your database applications with source code

Source code includes all db_VISTA libraries and utilities.

1. Recompile our run-time libraries utilizing non-standard compiler options.
2. Create a debugging library including a function traceback by activating pre-processor commands embedded in the source code.

Multi-user and LAN capability

Information often needs to be shared. db_VISTA has multi-user capability and supports simultaneous users in either multi-tasking or local area networking environments, allowing the same C applications to run under UNIX and MS-DOS.

Faster execution without data redundancy

Less data redundancy means reducing disk storage requirements and maximizing data access performance. A customer evaluating a leading competitor's product prior to purchasing db_VISTA benchmarked db_VISTA's retrieval time to be 276% faster than a leading competitor.

Complete documentation included

User manual contains 193 pages, 8 diagrams, 10 tables, appendices, an extensive index, plus a database application example. 9 chapters with complete instructions.

db_QUERY™ lets you ask more of your database

db_QUERY is a linkable, SQL-based ad hoc query and report writing facility. It's also royalty-free.

Circle no. 206 on reader service card.

30-Day Money-Back Guarantee

We wish to give you the opportunity to try db_VISTA for 30 days in your development environment and if not satisfied return it for a full refund.

Price Schedule

	db_VISTA	db_QUERY
Single-user	\$195	\$195
Single-user with Source	\$495	\$495
Multi-user	\$495	\$495
Multi-user with Source	\$990	\$990

FREE 60 Days Application Development Support
All software Not Copy Protected

Call Toll-Free Today!

To order or for information, call TOLL-FREE at 1-800-843-3313, then at the tone touch 700-992 or call 206-747-5570.

VISA and MASTERCARD Accepted

Read what others say about db_VISTA

"If you are looking for a sophisticated C programmers database, db_VISTA is it. In either a single or multi-user environment, db_VISTA lets you easily build complex databases with many interconnected record types. The multi-user implementation handles data efficiently with a LAN and Raima's customer support and documentation is excellent. Source code availability and a royalty-free run-time is a big plus."

*Dave Schmitt, President
Lattice, Inc.*

"Not 'yet another user-friendly database,' it is a DBMS aimed at the technical C programmer instead of the non-technical end-user."

*Hal Schoolcraft, Data Based Advisor
March, 1985*

"On the whole, I have found db_VISTA easy to use, very fast with a key find, and powerful enough for any DBMS use I can imagine on a microcomputer."

*Michael Wilson, Computer Language
September, 1985*

db_VISTA Version 2.11 Database Management System for C

Database Record and File Sizes

- ♦ Maximum record length limited only by accessible RAM
- ♦ Maximum records per file is 16,777,215
- ♦ No limit on number of records or set types
- ♦ Maximum file size limited only by available disk storage
- ♦ Maximum of 255 index and data files

Keys and Sets

- ♦ Key length maximum 246 bytes
- ♦ No limit on maximum number of key fields per record—any or all fields may be keys with the option of making each key unique or duplicate
- ♦ No limit on maximum number of fields per record, sets per database, or sort fields per set
- ♦ No limit on maximum number of member record types per set

Utilities

- ♦ Database definition language processor
- ♦ Interactive database access utility
- ♦ Database consistency check utility
- ♦ Database initialization utility
- ♦ Multi-user file locks clear utility
- ♦ Key file build utility
- ♦ Data field alignment check utility
- ♦ Database dictionary print utility
- ♦ Key file dump utility
- ♦ ASCII file import and export utility

Features

- ♦ Multi-user support allows flexibility to run on a local area network
- ♦ File structure is based on the B-tree indexing method and the network database model
- ♦ Run-time size, variable—will run in as little as 64K, recommended RAM size is 256K
- ♦ Transaction processing assures multi-user database consistency
- ♦ File locking support provides read and write locks on shared databases
- ♦ SQL-based db_QUERY is linkable
- ♦ File transfer utilities included for ASCII, dBASE optional

Operating System & Compiler Support

- ♦ Operating system's MS-DOS, PC-DOS, Unix, Xenix, Macintosh and Amiga
- ♦ C compiler's Lattice, Microsoft, DeSmet, Aztec, Computer Innovations, Xenix and Unix

Independent Benchmark Results

Eleven key retrieval tests on sequentially and randomly created key files. Benchmark procedure adapted from "Benchmarking Database Systems: A Systematic Approach" by Bitton, DeWitt, and Turbyfill, December, 1983.

Total Retrieval Time of 11 Tests
db_VISTA : 671.24
Leading Competitor : 1,856.43

RAIMA™
CORPORATION

12201 S.E. Tenth Street
Bellevue, WA 98005 USA
(206) 747-5570
Telex: 9103330300 BCN RIVERTON

1 (800) 843-3313

at the tone touch 700-992



pausing, until any key is pressed.

CR—(or Enter) Print next line. Like any other command, the CR command may be preceded by a count.

space—Print next screen.

Implementation

The code for more itself is in Listing One (page 64). Most of the external routines listed on lines 17–28 should be provided, in one form or another, with your compiler (these are for the Microsoft compiler). Exceptions are *b_getc()*, *look()*, and *filelength()*. The first two are in Listings Two and Three, page 78; I'll discuss them later. *Filelength()* is a Microsoft routine that returns the size of a file in bytes. If you don't have this routine, you can find the file length by seeking to the end of file and noting the file position returned by *fseek()*. An example is

shown in Table 2, below. The same techniques can be used with *open()* and *lseek()*. Because the file length is determined only once (on line 633), the penalty of using a seek isn't too great.

More keeps track of the starting position of each line on a stack. (A line's position is the offset, in characters, from the beginning of the file to the first character on the line.) Every time a line is input, the position of the first character is stacked. When you go backward a page, two pages' worth of these positions are popped off the stack, then a page is printed, starting at the last-popped line. Printing the page causes a page's worth of lines to be input with those line's positions getting restacked as part of the input process. The stack itself (*Stack*) and the stack pointer (*Sp*) are declared on lines 59–60. The macros on the following lines do various stack-maintenance tasks. *STACKFULL* evaluates to true if the stack is full, *STACKEMPTY* is true if the stack's empty, and *CLEAR*

_STACK deletes all items currently in the stack and resets the stack pointer. *TOS* evaluates to the entry at the current top of stack. The entry isn't popped, however. *BACKSCRN* evaluates to either the stack entry that's at one page's offset from the top of stack (that is, to the position of the top line on the screen) or to zero if there aren't that many lines on the screen (as will happen with a small file). It's used by the *o* command.

Various stack-maintenance subroutines are needed, too. *Push()* and *pop()* on lines 125–144 do what you'd think they would. *Comp_stack()* (lines 147–164) is called from *push()* when the stack is full. It compresses the stack by removing every other entry. This way you won't lose all the information on the stack in the event of an overflow; you'll just lose a little resolution when you go backward in the file with a *b* command. Note that the default stack size is 6K, so it's pretty unlikely that you'll run out of stack.

A help screen is printed by *help()* on lines 72–107. It uses the IBM box-drawing characters to put a box around the help message. You'll want to use dashes and vertical bars if you're not running the program on a PC.

The file position, represented as a percentage, is printed at every command prompt using the *percent()* routine on lines 258–264. Note that the cast is necessary here because, as *TOS* and *Flen* are both integral types, *TOS/Flen* evaluates to zero if the cast isn't present. The problem here is that expressions are evaluated two terms at a time, and the type of the temporary variable used to store intermediate results will be the same as the two operands. Because *TOS* and *Flen* are both *longs*, an integer division is done and the result will be truncated to zero and stored in a *long*. The compiler then evaluates the **100.00* part of the expression. Because the value of the temporary variable that's used to hold *TOS/Flen* is a *long* and *100.00* is a *double*, the *long* is promoted to *double* before the multiplication is performed. This promotion won't restore the fraction that was discarded in the initial division, though, so if the cast wasn't present, the expression would evaluate to zero.

Regular expression searching is

Usage: more [+<num>] [file. . .]

Print all files in list on the screen, pausing every 23 lines.

If + is specified, more will start printing at character <num>.

Stdin is used if no file is specified so more can be used at the end of a pipe. One of the following commands may be executed every time the program pauses:

bgo (B)ack a page
ego to end of file
ngo to (N)ext file
oprint (O)ffset from start of file in bytes
q(Q)uit (return to DOS)
s(S)kip one line (w/o printing)
r(R)ewind file (go back to beginning)
!execute a program (type blank line at prompt to
execute previous ! cmd)
/search for regular expression (type blank line at prompt
for last)
ESCscroll until any key is pressed
CRprint next line
SPprint next screen
anything elseprint list of legal commands

All commands may be preceded by a count.

Table 1: More's commands and command-line syntax

FILE	*fp;
int	length;
fp	= fopen("file", "r");
length	= fseek(fp, 0, 2); /* Offset 0 from EOF */
	fclose(fp);

Table 2: Finding a file length with *fseek()*

done by `search()` on lines 338-379. The subroutines `makepat()` and `matches()` aren't in the listing. They are the same routines as are used by `grep`. (See the Availability section.)

The `!` command is processed by `execute()` on lines 383-444. Note that I'm using a `spawnlp()` call rather than a `system()` call to create the child process. This means that you can't execute a batch file directly from within more, though you can do it indirectly, as I explained earlier. If you're going to execute a lot of batch files, you may want to change line 432 into a `system()` call.

Note that a bug in Microsoft C, Version 3.0, forces you to close the current input file (on line 591) before calling `execute()` and then reopen it on returning (lines 595-605). This will cause problems if you're getting input from standard input because you won't be able to reopen the file, not having a file name to give `fopen()`. Consequently, if you're likely to use the `!` command, you'll have to create a temporary file rather than use a pipe.

The remainder of the program is pretty much self-explanatory. The one other pipe-related problem is solved with the `b_getc()` and `look()` functions in Listings Two and Three. You have to get commands using ROM BIOS input routines because you might be using standard input for the file being printed. `B_getc()` gets a character from the BIOS, using the keyboard interrupt (0x16). `Look()` is a keyboard look-ahead function. It returns 0 if no key has been typed. Otherwise it returns the key in the scan-code/character-code format used by the BIOS. You can find more information in the *DOS Technical Reference* (buried in the BIOS listing as a comment) and in *The Peter Norton Programmer's Guide to the IBM PC*. Although `look()` was written in assembler for speed reasons, it's pretty easy to move it to C. Most of the file is just overhead for the Microsoft compiler. The actual work is done on lines 37-41, and the return value is in `ax`.

Availability

The pattern-matching routines were originally published as part of `grep` in *DDJ*, October 1984. Back issues are available for \$5 from *DDJ*. `Grep` is also available electronically as part of the

/util program package listed in the *DDJ* catalog ad (page 105) and is included in both *Dr. Dobb's Toolbook of C* and in *Dr. Dobb's Bound Volume 9*.

All the code printed in this month's issue is available on CompuServe in DL 1 (type `ddjforum`). The entire program, along with the pattern-matching routines and an executable version, is also available on an IBM PC-compatible disk for \$25 from Software Engineering Consultants, P.O. Box 5679, Berkeley, CA 94705.

Erratum

Ron Albury found a bug in Listing Seven of my AVL tree routines (August 1986) that could cause the insert function to fail on an attempt to insert a conflicting node. To fix the problem, add the line `h = 0`; immediately below line 68 (page 92). **DDJ**

(Listings begin on page 64.)

Vote for your favorite feature/article.
Circle Reader Service No. 2.

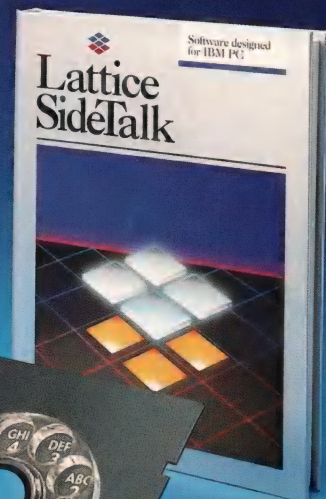
SideTalk™

It minds the phone while you do your work

No more interruptions! Because now SideTalk can transfer your files or receive your mail while you're using your computer for other important matters. And SideTalk is programmable with its own BASIC-like language. What's more, you're never more than a keystroke away from all of the power of SideTalk.

When we say SideTalk is the best telecommunications program on the market, we're not just talking out of the side of our mouth.


Lattice
To place your order:
Lattice, Incorporated
Post Office Box 3072
Glen Ellyn, Illinois 60138
312-858-7950
TWX 910-291-2190



Only
\$119.95



Circle no. 101 on reader service card.

SOFTWARE
COMPONENTS
GROUP



MOTOROLA

DIGITAL
RESEARCH

GEM™/GEM™ DOS 68K

RM/COBOL™

Pascal 2

Oregon
Software

RTUX

DIGITAL
RESEARCH

Concurrent™ DOS

Technical
Documentation
UNIX
SYSTEM V/68



MOTOROLA
MICROSYSTEMS
SYSTEM V/68
Volume IV

polyFORTH II

VRTX/68000 USER'S GUIDE

PL/M

ELESOFT **ADA**

FORTRAN 77

Technical
Documentation



MOTOROLA
MICROSYSTEMS
VERSAdos
Volume I

FRANZ LISP
Reference Manual

VERSAdos

BASIC



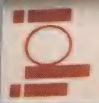
Whitesmiths, Ltd.



MPROLOG

MODULA-2

Graphic Kernel System **GK-2**



STOS

MACRO ASSEMBLER 68000/68010/68020

Put your chips on the software.

The best bet.

If you're betting your future on a 32-bit microprocessor-based system, you'd rather bet a winner. Today, the clear winner is the MC68020 microprocessor. It's faster. It's fully supported in hardware and software. It's backed by what others have called, "The strongest support team in the business." And, we're delivering production quantities. Now.

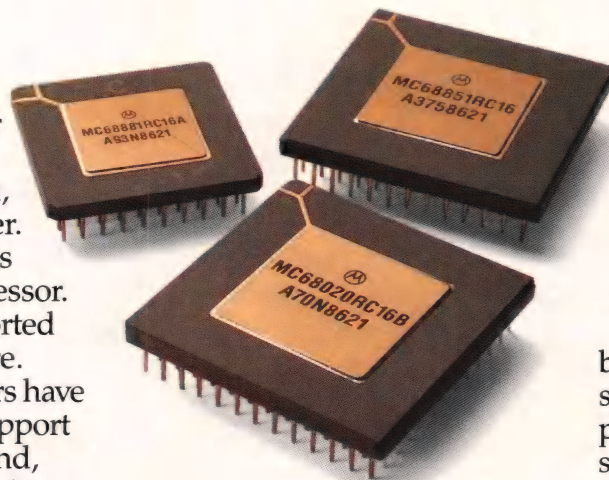
In fact, last year Motorola shipped over 50,000 units and this year we expect to ship more than a quarter of a million MC68020s.

Software support. Now.

When you choose the MC68020, you get the largest body of software available for 32-bit microprocessor-based machines. Operating systems. Languages. Tools and utilities. Host independent development systems. And, of course, applications. In fact, 80% of the microprocessor-based system designs running UNIX® OSs are based on M68000 Family chips.

The fastest just got faster.

At 25 MHz, the latest version of the MC68020 and its earlier 20 MHz version remain



The MC68020

the fastest general-purpose processors on the market today. And fully compatible. But speed alone isn't the answer. And for the generation beyond the MC68020, we're committed to higher performance devices while maintaining the M68000 Family software base.

Chips, boards or boxes.

Motorola is with you at every level. As the world's leader in VMEbus products, supplying MC68020-based VMEmodule™ and VMEsystem™ products, we can work with your development team at virtually every level, including technical training.

Betting the winner.

There's a lot of 32-bit talk in the press. Sampling quantities, blue sky and vaporware are fine, but maybe you'd rather not bet your company on them.

When you're planning to build the kind of leading-edge systems that win in the marketplace, Motorola offers you speed to market, aggressive pricing and the strength of a fully-realized 32-bit environment.

If you're an executive evaluating microprocessors, software and support, call us at 1-800-521-6274. We'll arrange a one-on-one management-level meeting. Fast. And we'll show you why a better microprocessor, coupled with here-and-now volume production, software and support, adds up to a winning combination.

UNIX is a registered trademark of AT&T
Concurrent DOS and GEMDOS is a trademark of Digital Research Inc.
VERSAdos is a registered trademark of Motorola Inc.
VRTX is a registered trademark of Hunter & Ready, Inc.
RTUX is a trademark of Emerge Systems
ADA is a trademark of the Department of Defense
VMEmodule and VMEsystem are trademarks of Motorola Inc.

**MOTOROLA**

Programming on the 80386

by Ross Nelson

Intel recently introduced the 80386, its entry into the 32-bit microprocessor derby. The 80386 can run all programs developed for the 80286, which in turn runs programs designed for the 8086 and 8088.

Because of this compatibility with its widely used predecessors, the 80386 is likely to be very popular. The 80386 (or 386 for short) is not merely bigger and faster, however; Intel has made some significant architectural changes as well. The enhancements that I'll examine in this article include the elimination of the 64K segment restriction; enhanced instruction set and operand addressing; the ability to run 32-bit and 16-bit software simultaneously; and virtual memory support, including paging.

Operating Modes

Like its predecessor the 286, the 386 operates in either real address mode or protected virtual address mode, usually just called real mode and protected mode, respectively. In real mode, the 386 is practically indistinguishable from an 8088 or an 8086. Real mode carries with it all the restrictions of the 8086—most important, only 1 megabyte of memory is directly addressable. As in the 8086, physical addresses are created by multiplying the segment register value by 16 and adding an offset.

Object-code compatible with the 286, the 386 also operates in protected mode. Unlike the 286, however, it also performs 32-bit operations. All the architectural enhancements of the 386 are available when running 32-bit instructions in protected mode. This is the way the processor was designed to run and is therefore called its *native mode*. The other modes (real mode, 16-bit protected

The 80386 is likely to be very popular because of its compatibility with the 8086, 8088, and 80286.

mode, and virtual 8086 mode) are called *emulation modes*. The basic protection mechanism of the 386 is identical to that of the 286. It is outside the scope of this article to describe the full protection model of the

286 and 386; therefore, I will essentially ignore gate descriptors, tasking, and privilege levels. A short review is provided in the inset entitled "The 286/386 Protection Model" on page 39. The extensions to the 386 are primarily in memory addressing, so it's appropriate to review protected mode addressing in the 286.

In protected mode, there is a dramatic change in the way the processor behaves. In real mode, the program currently executing interprets memory values, and the processor is merely the vehicle for manipulation of the data provided by the program. In protected mode, however, the processor assigns semantic meaning to certain blocks of memory independently of whatever program may be running. Each block of memory the processor recognizes I call a *system object*. The most common system object is the *descriptor*. Other objects include *descriptor tables*, which contain descriptors; *segments*, which are blocks of memory; and *gates*, which restrict access to segments and help enforce the protection rules. Each segment, gate, and table has an 8-byte descriptor that defines it. Descriptors contain information about the object, such as its size, type, location in memory, and protection attributes. Figure 1, page 34, shows a typical segment descriptor for the 286.

In protected mode, memory is accessed via a descriptor. The contents of segment registers do not point to specific memory paragraphs but are treated as indices into descriptor tables. The processor requires the existence of a global descriptor table (GDT) and an interrupt descriptor

Ross Nelson, 1821 Ashmeade Ct., San Jose, CA 95125

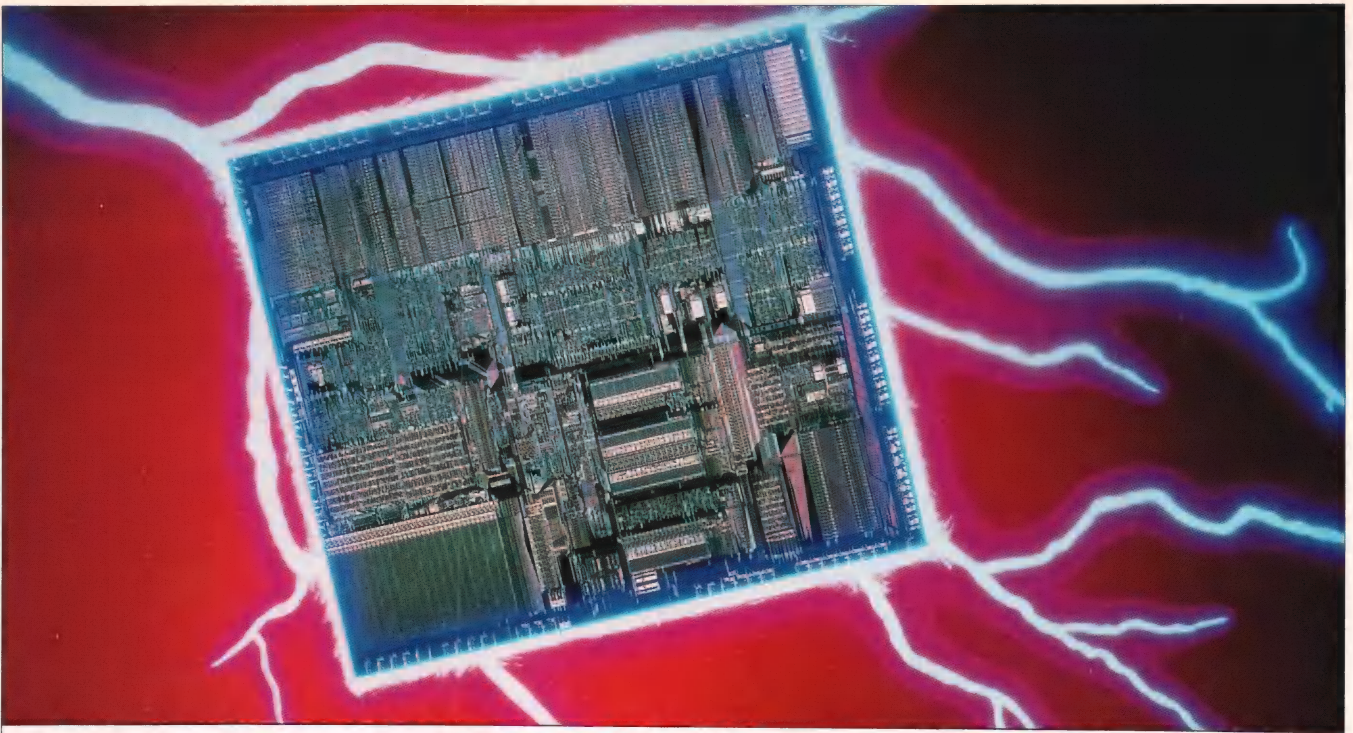


table (IDT). It also allows optional local descriptor tables (LDTs) to be present. The *GDTR* and *IDTR* registers point to the GDT and IDT, respectively. All other system objects, including segments, LDTs, and gates, are pointed to by descriptors. Figure 2, page 34, outlines the hierarchy of pointers to system objects.

When a memory reference instruction, such as *MOV AX, [200]*, is executed, the base address from the descriptor selected by the *DS* register is added to the offset from the instruction (in this case 0200H) to generate the *linear address*. In the 286, the linear address becomes the physical address that goes out over the processor bus. In the 386, however, the linear address passes through the paging mechanism, which generates the final physical address. I will examine paging a little later. First, I'll look at how descriptors have been changed in the 386.

In the 286, the last 2 bytes of the descriptor must be 0. In the 386, though, these bytes can take on other values. Figure 3, page 34, shows the fields in the last 16 bits of a descriptor on the 386. Eight of the bits have been used to extend the linear address space to 32 bits, another 4 bits go toward extending the limit field, 2 bits are used as flags, and another 2 bits are reserved for some future processor. At first glance, these extensions grant you a physical address space of 2^{32} , as expected, with a maximum segment size of 2^{20} , or 1 megabyte.

Have you been saddled with a new segment limitation? Fortunately, no. The *G* bit stands for segment granularity. When reset to 0, as in 286 code, the size of a segment (as indicated by the limit field) is measured in bytes. But when the *G* bit is set to 1, the segment size is measured in *pages*. Each page is 2^{12} bytes (4K) long. Therefore, the maximum segment size is 2^{20} pages times 2^{12} bytes per page, or 2^{32} bytes. The *D* bit, which Intel calls the *default bit*, is active only in executable segment descriptors. When set to 1, it means that the native mode, 32-bit instruction set is to be used. When reset, the processor interprets opcodes as if it were a 286.

Native Architecture and Instruction Set

The 386 microprocessor holds 34 different registers, grouped into four classes. These registers are illustrated in Figure 4, page 34. The first group, general-purpose registers, are the ones most commonly dealt with. They act as accumulators and index registers, and their names are derived from the corresponding registers of the previous generations of processors.

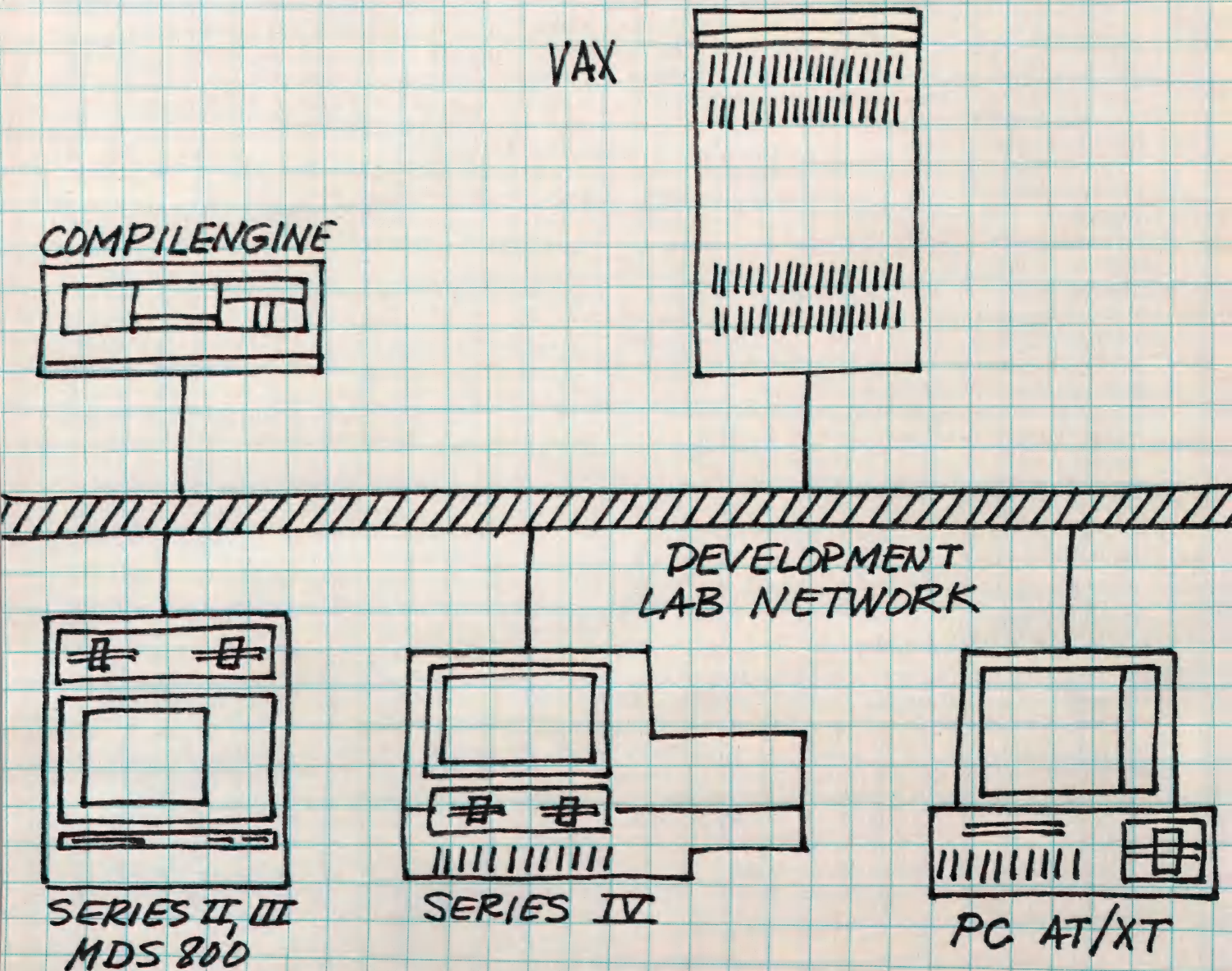
The next register class, segmentation and protection, is also familiar from the 286, although there are two new entries. In addition to the *SS*, *CS*, *DS*, and *ES* segment registers, there are the *FS* and *GS*. These segment registers are used only when the special segment override prefixes *FS:* and *GS:* are found in the instruction stream. Associated with most of the registers in this class is a special on-chip cache that holds the descriptor information associated with each segment. This precludes the necessity of reading the descriptor table every time an access to a given segment occurs.

The registers in the control class are partially familiar. *EIP* is the 32-bit extension of the instruction pointer, and *EFLAGS* contains two additional flag bits. The control registers (*CR0*—*CR7*) are new to the 386, but *CR0* contains what was called the machine status word (MSW) on the 286. These control registers contain information necessary for controlling processor extensions (80387) and paging.

The final class of registers, test and debug, are completely new. The debug registers (*DR0*—*DR7*) allow a software debugger to set the kind of breakpoints that used to require an expensive emulator to generate. I'll describe use of these registers in more detail later. The test registers, *TR6* and *TR7*, are used to verify that the paging cache is working correctly.

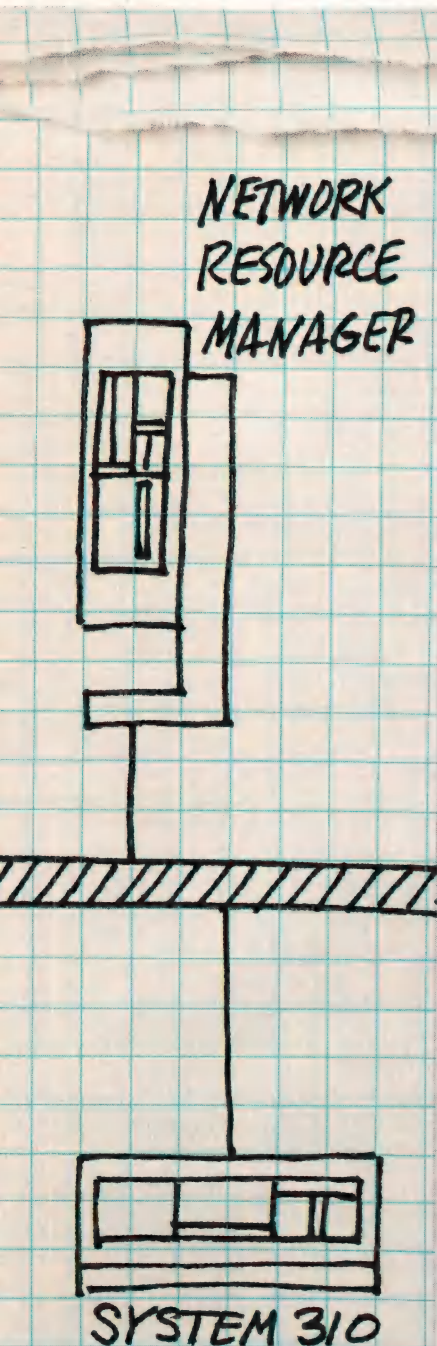
In addition to expanding the address space and register set of the processor, Intel has also expanded the addressing modes of the instruction set. The instruction format of the previous generations of processors was an opcode byte, followed by the *modr/m* byte, followed by any

NOBODY ELSE HAS MADE THE CONNECTION.



There's only one way to link the extensive resources of Intel's microprocessor development system with the power of a DEC VAX® and the affordability of the IBM® PC.

And that's via our enhanced version



of OpenNET™ for the development lab.

Thanks to this open architecture network, engineers can have immediate, and transparent, access to other team members' work.

Plus you have the ability to add special-purpose hardware to make those teams more productive.

For example, by connecting our high-performance file server, the Network Resource Manager, you can off-load file management and job distribution from the shoulders of your design team. On the off chance nobody wants to spend his time chasing down floppy disks.

And then you can hook up an 80286-based Compilengine to off-load compute-bound compilations from VAXs and workstations. Leaving them, and their human partners, more time for interactive tasks.

Equally important, the network maximizes the value of your existing development

hardware while minimizing your outlay for new equipment.

That's because OpenNET adheres very rigidly to some very flexible standards. Standards like Ethernet/IEEE 802.3. And the ISO message delivery and Intel/IBM/Microsoft® Network File Access protocols.

All of which means existing development hosts, languages and tools, including ICE, are instantly compatible with the latest ones. So you can avoid obsoleting one set of tools just to use another.

We'll even take full responsibility for servicing and supporting your network. Anywhere in the world.

Sound like we've got things together? Then call us at (800) 548-4725.

Ask to meet with one of our experienced network engineers. We'll see you make all the right connections.

intel®

VAX is a registered trademark of Digital Equipment Corporation. IBM is a registered trademark of International Business Machines Corp. OpenNET is a trademark of Intel Corporation. Microsoft is a registered trademark of Microsoft Corporation. © 1986 Intel Corporation.

operands required. Because only a limited number of addressing modes could be encoded in the *modr/m* byte, certain registers took on dedicated functions. Only *BX*, *SI*, *DI*, and *BP* could be used for indexing or indirection and only in certain combinations. These restrictions have been greatly relaxed in the 386 with the addition of another address mode byte following the *modr/m*.

This new byte, called *s-i-b*, for scale-index-base, extends the addressing capabilities of the 386 in two significant ways. First, it allows any of the eight general-purpose registers to be used as base or index registers, in any combination. This makes the job of compiler writers much easier because they no longer have to worry about having the results of address computations in the proper register. Any of the registers is proper.

In addition, the scale portion of the *s-i-b* byte can be used to eliminate array index computation altogether in some cases. As an example, assume that array *FOO* contains several 32-bit floating-point numbers. The instruction sequence generated by the high-level language statement *SQRT(FOO[I+3])* is shown for the 8086, the 286, and the 386 in Table 1, below. Automatic scaling is allowed only for arrays whose elements are 2, 4, or 8 bytes long.

In the 8086 and 286 instruction sets, the most common operations affected either byte or word operands. The same is true of the 386 except that the *D* bit of the executing code segment is checked to see if the machine word is 32 bits long (*D*=1) or 16 bits long (*D*=0). The *MOVSW* instruction, for example, will copy a 16-bit quantity when *D* is 0 and a 32-bit quantity when *D* is 1. To allow a program running in the native (32-bit) mode to access a 16-bit quantity, an override instruction (066H) is provided, which tog-

gles the default operand size for the next instruction. While running in 16-bit mode, this opcode has the inverse effect—it allows access to 32-bit registers and memory operands.

The instruction repertoire of the 386 has been enhanced as well. Opcodes have been added to allow access to the new control, breakpoint, and debug registers, and new conditional and bit operators have been added. There are now double-precision shift operators and move byte with sign extension or zero extension. Table 2, below, lists the new mnemonics.

Paging

Paging has long been the most popular method of implementing virtual memory. Although virtual memory can be achieved with segmentation alone (as in the 286), paging methods are usually faster and simpler because the fixed page size maps easily onto the fixed sector sizes of disks, the most common secondary storage medium. The page size of the memory management unit (MMU) of the 386 is 2^{12} bytes, or 4K. It is no coincidence that the granularity bit of the segment descriptors deals with pages of the same size.

The low-order 12 bits are reserved to address within a page, leaving 20 of the 32 physical address bits to select the page. The additional 20 bits could be used as an index into an array of linear (virtual) to physical addresses, but this would require a table of more than 1 million entries (2^{20}) to be in memory constantly for each task. Instead, the upper 20 bits are divided into two 10-bit numbers. The highest order value is used to select one of 1,024 (2^{10}) page table directories. Each directory entry points to a page table containing 1,024 physical page addresses. The advantage of using this method is that only the directory entries must be guaranteed to be in memory at all times, whereas the page tables themselves may be swapped out to save working storage space. Note that, with 1,024 entries of 32 bits each, a page table is 4K long.

One of the control registers (*CR3*) points to the starting location of the page table directories. A copy of *CR3* is

8086	286	386
MOV AX, I	MOV AX, I	MOV EAX, I
ADD AX, 3	ADD AX, 3	ADD EAX, 3
MOV BX, AX	MOV BX, AX	FLD FOO[EAX * 4]
MOV CL, 2	SHR BX, 2	FSQRT
SHR BX, CL	FLD [BX]	
FLD [BX]	FSQRT	
FSQRT		

Table 1: Implementation of *SQRT(FOO[I + 3])* on 8086/286/386

MOVX	move byte to word, sign extended
MOVZX	move byte to word, zero extended
LFS, LGS	load pointer, new segment register
SHLD, SHRD	double word shift
BT	bit test
BTC	bit test and complement
BTS	bit test and set
BTR	bit test and reset
BSF	bit scan forward
BSR	bit scan reverse
SETcc	set byte if condition code (cc same as Jcc in conditional jumps)

Table 2: Instruction set additions for the 386

80386 Development Tools

In addition to the standard Intel development tools that have been available for some time, new products (both software and hardware) that can significantly speed 386 development tasks are starting to appear.

One such product is called the 386 Translator. It's a plug-in piggyback card that replaces the 80286 in a standard IBM PC/AT with an 80386 and some support circuitry. The new board allows developers to create software that takes advantage of the 386's ability to run simultaneously in several different modes. The only penalty seems to be that an AT with the 386 Translator board runs about 10 percent slower than an unmodified machine because of the wait states that must be inserted for 386 memory accesses.

The 386 Translator is available from American Computer & Peripheral Inc., 2720 Croddy Way, Santa Ana, CA 92704; (714) 545-2004.

A MEGABYTE FOR DOS!

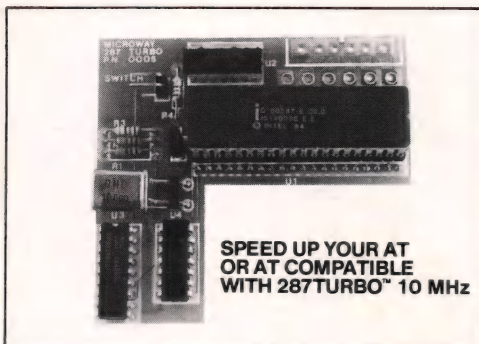
MicroWay is the world's leading retailer of 8087s and high performance PC upgrades. We stock a complete selection of 8087s that run from 5 to 12 MHz. All of our coprocessors are shipped with a diagnostic disk and the best warranty in the business - one year! We also offer daughterboards for socketless computers (NEC PC) and 287Turbo which increases the clock speed of the 80287 from 4 to 10 MHz. Our NUMBER SMASHER/ECM™ runs at 12 MHz with a megabyte of RAM and achieves a throughput of .1 megaflops with 87BASIC/INLINE, Intel For-

tran, or Microsoft Fortran. Software reviewers consistently cite MicroWay software and 8087 expertise as the best in the industry! Our customers frequently write to thank us for recommending the correct software and hardware to meet their specific needs. They also thank us for our same-day shipping! In addition to our own products which support the 8087 and 80287, we stock the largest supply of specialized software available. For more information call us at

617-746-7341

NUMBER SMASHER/ECM™ THE FASTEST ACCELERATOR CARD AVAILABLE

gives you 12 MHz speed in two modes: 704K or one megabyte of "Extended Conventional Memory." MEGASWITCH MMU and MegaDOS software make it possible to run DOS applications with up to 1015K using PC compilers, AutoCAD and Lotus 1-2-3. Does not require EMS software. Totally compatible. Priced from \$599 with 512K to \$1199 for complete package. Optional 8087-12 ... \$295



MicroWay® 8087 Support

For the IBM PC, PC XT, PC AT and Compatibles.

A2D-160™ MicroWay's Data Acquisition Board performs 160,000 12 bit Analog to Digital conversions per second! Includes software drivers. The fastest 12 bit A to D board available. For the IBM PC XT and compatibles. \$1295

87SFL™ MicroWay's Scientific Function Library contains 170 scientific and engineering functions. Callable from most 8087 compatible compilers. First Language \$250; Additional \$100.

MATRIXPAK™ manages a MEGABYTE! Written in assembly language, our runtime package accurately manipulates large matrices at very fast speeds. Includes matrix inversion and the solution of simultaneous linear equations. Callable from RM or MS Fortran, MS Assembler, or 87BASIC/INLINE. each \$99

87FFT™ Written in assembly language, performs Forward and Inverse FFTs on real and complex arrays which occupy up to 512 Kbytes of RAM. Also does convolutions, auto correlations, hamming, complex vector multiplication, and complex to radial conversions. Callable from most 8087 compatible compilers. \$200

87FFT-2™ performs two-dimensional FFTs. Ideal for image processing. Requires 87FFT \$100

FASTBREAK™ employs the 8087 to increase the speed of Lotus 1-2-3™ Version 1A or 1A* by up to 36:1 \$79

87Verify™ For users who have to be absolutely sure of their results! This background task periodically performs an 8087 accuracy and stress test. \$49

Microsoft Fortran V 3.31 \$209
IBM Professional Fortran \$565
Ryan-McFarland Fortran V 2.0 \$399
NAG Fortran Library \$300
Grafmatic for Fortran or Pascal. \$125
MultiHalo Graphics (1 language) \$189
LABTECH NOTEBOOK \$745
UnkelScope \$549
INTEL ABOVE BOARD CALL
JRAM, AST, MAYNARD CALL

MegaPage™ The only Intel-Lotus EMS board which comes with two megabytes of cool-running, low power drain CMOS RAM installed. Includes RAM disk, print spooler, disk cache, and EMS drivers. For the IBM PC, XT & compatibles. \$549

MegaPageAT/ECC™ EMS card for the PC AT and compatibles includes Error Correction Circuitry. With ECC, 11 RAM chips cover 256K so the user never encounters RAM errors. With 1 megabyte CMOS \$799; with 3 megabytes CMOS \$1295. Optional serial/parallel daughterboard. \$95.

DFixer™ Our disk utility which thoroughly checks PC or AT hard disks for bad sectors and updates the MS DOS file allocation table accordingly. Solves the AT hard disk problem! \$149

DOptimizer™ Optimizes the way your hard disk or floppy stores its files. Speeds up accesses by recombining fragmented files. \$49

DCache™ Our disk caching software speeds up your I/O by storing repetitively used tracks in memory. The amount of memory used can be selected in 64 Kbyte banks. \$49

87MACRO/DEBUG™ Contains all the pieces needed for writing 8087/80287 assembly code & MicroWay's 87DEBUG debugger. \$199

OBJ→ASM™ A multipass object module translator and disassembler. Produces assembly language listings which include public symbols, external symbols and labels commented with cross references. Ideal for patching object modules for which source is not available. \$200

87BASIC™ includes patches to the IBM BASIC or MS Quick BASIC Compiler for USER TRANSPARENT 8087 support. Provides super fast performance for all numeric operations including trigonometrics, transcendental, addition, subtraction, multiplication, and division. each \$150

87BASIC/INLINE™ converts the output of the IBM BASIC Compiler into optimized 8087 inline code which executes up to seven times faster than 87BASIC. Supports separately compiled inline subroutines which are located in their own segments and can contain up to 64 Kbytes of code. This allows programs greater than 128K! Requires the IBM BASIC Compiler Version 1 and a Macro Assembler. Includes 87BASIC ... \$200

MICROWAY UDI runs RTOS or RMX compilers under DOS. \$300

8087 UPGRADES

All MicroWay 8087s include a one year warranty, complete MicroWay Test Program and accurate installation instructions.

8087 5 MHz..... \$109
For the IBM PC, XT and compatibles.

8087-2 8 MHz..... \$149
For Wang, AT&T, DeskPro, NEC, Leading Edge.

80287-3 5 MHz..... \$179
For the IBM PC AT and 286 compatibles.

80287-6 6 MHz..... \$229
For 8 MHz AT compatibles.

80287-8 8 MHz..... \$295
For the 8 MHz 80286 accelerator cards.

NEC V20, V30..... \$16, \$30

64K RAM Set 150ns..... \$10

256K RAM Set 150ns..... \$29

256K RAM Set 120ns..... \$39

128K RAM Set PC AT..... \$49

287Turbo™ 10 MHz If you own an AT, Deskpro 286 or AT compatible, this is the card you need to get reasonable numeric performance. It plugs into your 80287 socket and includes a specially driven 10 MHz 80287. The card comes in three configurations. The IBM AT version includes a hardware RESET button \$450

287Turbo 8 MHz..... \$369

87/88Turbo™ is a stubby card which includes a clock calendar and a speed controller which changes the speed of your motherboard from 4.77 to 7.4 MHz. Its use requires your PC to have a socketed 8284. Typical speed increase is 1.6 to 2.0. The card overcomes slow hardware by slowing up only when such devices are accessed and running at full speed otherwise \$149
Optional 8087-2 \$149

286TurboCache™ This new MicroWay accelerator uses 8K of cache memory and 80286/80287 processors to provide an average speed increase of 3:1 for most programs. Call for specifications and benchmarks \$595

Call for our complete catalog of software which supports the 8087.
In London, please phone 223-7762

MicroWay

P.O. Box 79
Kingston, Mass.
02364 USA
(617) 746-7341

(continued from page 32)

stored for each task in 386 native mode. Figure 5, page 38, illustrates translation from linear to physical address. Because the directory pointers and the page table pointer both reference a 4K page, only 20 bits of the 32-bit word are used as a page address by the MMU. This frees up the 12 low-order bits for other uses. The lowest-order bit (called the *P* bit) is used to mark whether the page is actually present in physical memory. If a memory reference

occurs and either the page table or the physical page is marked "not present," a page fault (*int 14*) occurs, and the operating system is responsible for reading the page into physical memory. The other 11 bits have various uses; some are used by the hardware to mark whether pages have been used and to provide a simple user/supervisor protection scheme, and three of the bits can be used by the operating system. Note that whenever the *P* bit is 0 or "not present," the pointer does not contain a physical memory address and the operating system can use the other 31 bits as it chooses—typically to hold the disk sec-

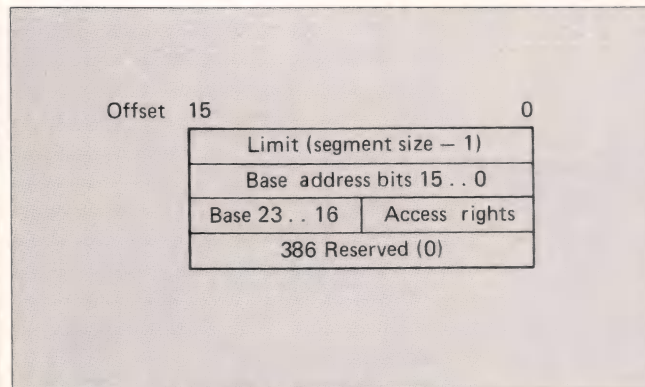


Figure 1: 286 segment descriptor

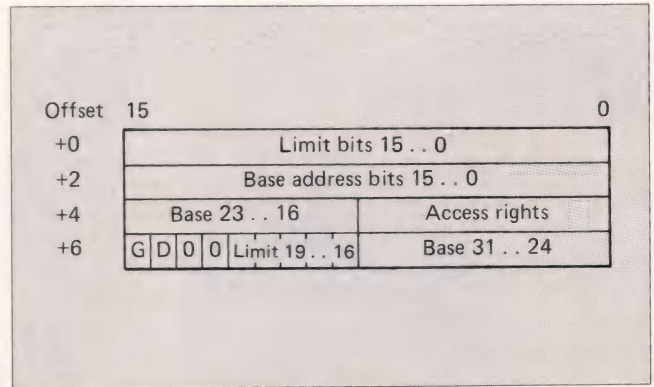


Figure 3: 386 segment descriptor

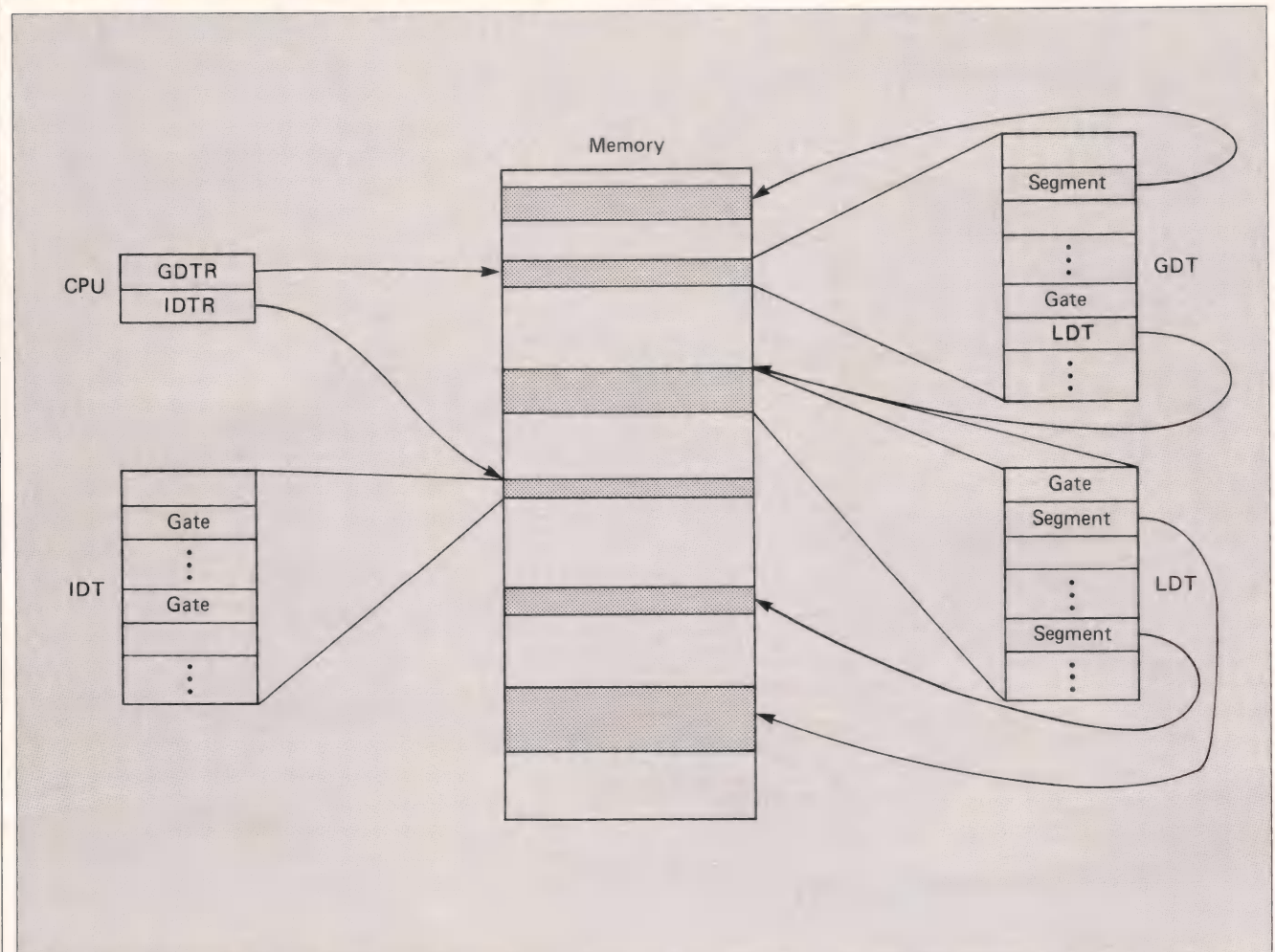


Figure 2: Descriptor hierarchy

tor that contains the primary storage memory image.

From this discussion, it would seem that every memory-reference instruction executed while paging was enabled would actually require three memory fetches to complete: the first to fetch the page directory, the next to fetch the page table, and the third to finally read the operand itself. To prevent this slowdown, the 386 contains a cache, called the translation lookaside buffer (TLB), which holds the 32 most commonly referenced page table entries. If a cache "hit" occurs, no lookup penalty will be exacted. Intel estimates that only 2 percent of address lookups will require the three-stage memory references.

As a further performance optimization, the MMU is on-chip. Microprocessor systems with an external MMU often require delays equivalent to one wait state while the MMU determines whether a page fault has occurred.

Performance

Instruction durations for the 386 are measured by the number of clock cycles required for an instruction to complete. When coupled with the processor clock rate, an instruction time (in nanoseconds or microseconds) can be generated.

For the most part, the number of clocks required for a given instruction on the 386 is the same as it was on the 286. When the 286 was first available, however, it ran at clock rates of 6 and 8 MHz. The 386 can run at speeds of

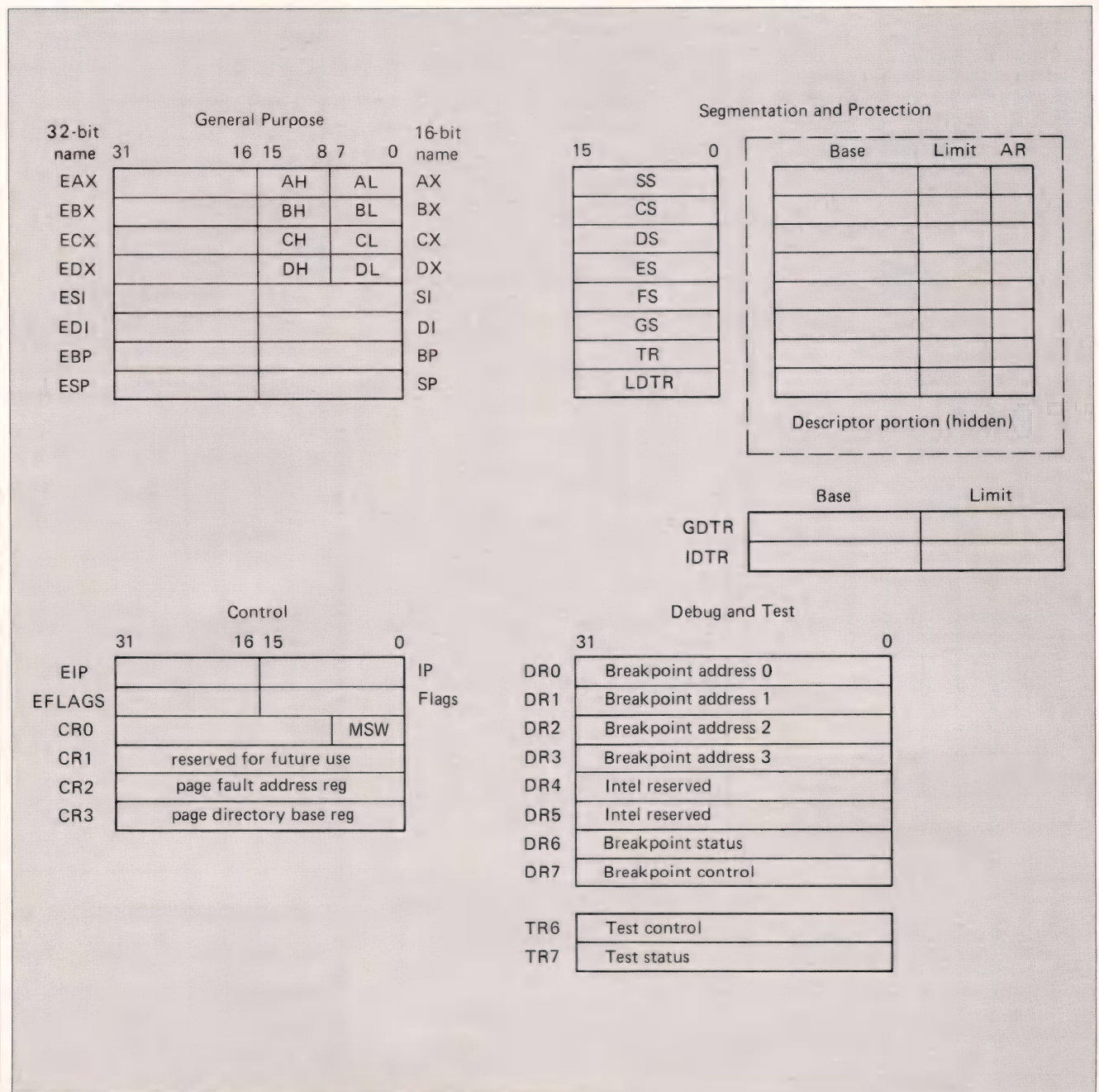


Figure 4: 386 CPU registers

Block software theft...

with
THE BLOCK.™
Patented protection for your profits.

Software-based security systems can be cracked too easily.

But now there's THE BLOCK,™ an inexpensive device you supply with your software and without which it will not function.

Users simply plug THE BLOCK into an external port of their PC, where it is totally transparent. However, commands you program into your software check for the presence of the device before allowing your program to run.

What's more, THE BLOCK has the added protection of a U.S. Government patent to prevent mass distribution of illegal "work-alikes."

THE BLOCK also lowers the price of security. Prices range from \$39 for single units to \$24 for quantities over 1,000.

Call for your demonstration unit today...

So give your product investment the unmatched protection of THE BLOCK. Send today for complete information and a demonstration unit—or to get it even faster, phone us at (203) 329-8870.

And block those hackers from hacking holes in your software profit plans.



Software Security, Inc., 870 High Ridge Rd., Stamford, CT 06905.

Circle no. 170 on reader service card.



It's good for your system.

FULLY Integrated, Data Entry Windows!

- Complete input formatting
- Unlimited Validation
- Full attribute control
- Multiple virtual windows
- Fully automatic, collision proof overlay and restore
- Print to & scroll background windows
- Animated window "zoom"
- Move, grow, shrink, hide, or show any window
- "Loop function" allows processing while awaiting input

AND MUCH MORE!

\$149.95

Includes 100% source, tutorial, reference manual, examples, and sample programs.

Specify Microsoft, Lattice v2 or v3, Computer Innovations, Aztec, DeSmet, or Mark Williams. Ask about Unix.

100% Money Back Guarantee

NOW ... VCScreen!

Our new interactive screen "painter" actually lets you draw your data entry windows! Define fields, text, boxes & borders. Move them around. Change attributes. Then the touch of a button generates C source code calls to the Vitamin C routines!

\$99.95

Requires Vitamin C Library above. For IBM & compatibles.

**For Orders Or Information,
(214) 245-6090**

**Creative Programming Consultants, Inc.
Box 112097 Carrollton, TX 75011-2097**

Include \$3 ground, \$6 air, \$15 overnight shipping, \$25 if outside USA. Texans add 6 1/2% tax. All funds must be in U.S. dollars drawn on a U.S. bank.

creative
PROGRAMMING

12.5 and 16 MHz, approximately twice as fast. It must be emphasized that clock rate alone is an inadequate measure of performance. For one thing, the bandwidth of the 386 is also twice that of the 286; that is, each 386 instruction is capable of processing 32 bits of data, whereas the 286 processes only 16 bits. Programs written in high-level languages that are recompiled for native mode can also see performance gains based on the use of new instructions and addressing modes.

As a general rule, you can assume that programs running in emulation mode will see a performance gain equivalent to the change in clock rate between the two computers being compared. Assuming you are comparing a 6-MHz 286 with a 12-MHz 386, this means a speedup of two times, all other things being equal. Comparing a 286 program against a 386 native mode program, two times should be the minimum performance improvement. Depending on the application, gains of four to five times are easily attainable.

Additional Features

One unique feature of the 386 architecture is its support for the programmer. In addition to the single-step interrupt (*int 1*) and software breakpoint interrupt (*int 3*), also found on the 8086 and 286, the 386 provides four breakpoint registers that can be set to match on instruction execution, data accessed, or data written for a given address. This feature allows debuggers written for the 386 to implement commands such as *GO til variable ZOT modified*.

Another selling point for the processor is the virtual 8086 mode. With this option, an operating system that runs in the native mode (Unix, for example) would be able to run MS-DOS and DOS applications as subtasks. Because programs running in this mode actually generate the same linear addresses as an 8086 (0-1 megabytes), this option is useful only in an operating system running with paging enabled so that the addresses of multiple DOS tasks can be mapped to their different physical memory locations.

Also, for those who cannot abide segmentation of any sort, the 386 can masquerade as a linear-address-space

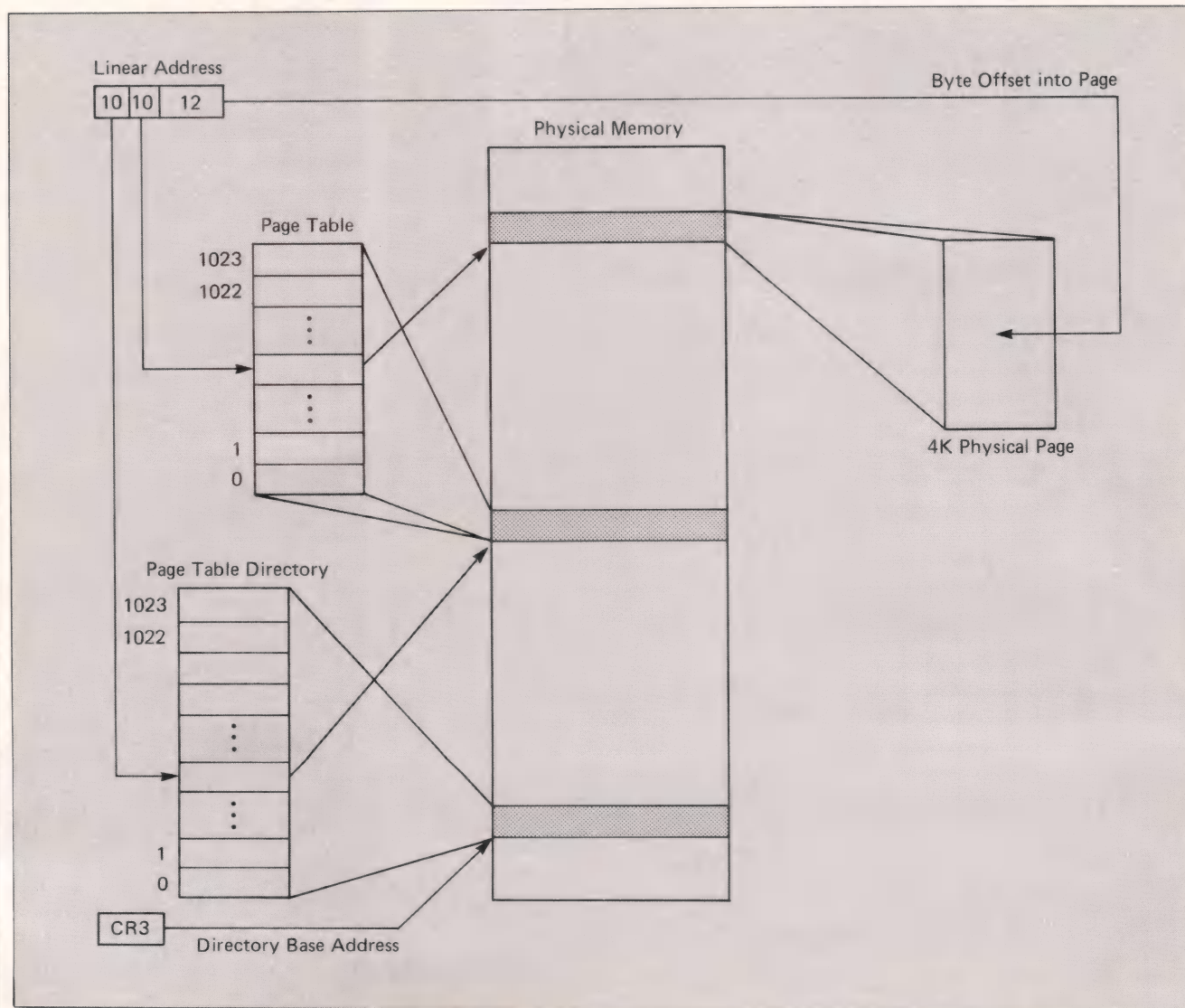


Figure 5: Linear-to-physical-address translation

The 286/386 Protection Model

Intel's protected mode processors, the 80286 and 80386, use a variety of methods to safeguard the security of data belonging to one process from corruption by another process. By making all memory references indirect (through a descriptor), the CPU can verify that the operations specified are valid.

The access rights (AR) byte of a descriptor contains the key to the operations that are legal for the object. Bits in the AR byte specify whether the object is currently present in memory, the privilege level of the object, and the type of object.

When a program is running, the privilege level of the code segment is set to the process's current privilege level (CPL). There are four privilege levels, ranging from level 0 (most privileged) to level 3 (least privileged). No process is allowed to access an object more privileged than itself. Typically, the operating system will consist of segments of higher privilege than applications, and it will therefore be protected from accidental or malicious attempts at modification.

In the interests of efficiency, certain routines are often shared between processes. I/O libraries, for example, are often part of the operating system but are usable by all applications. The protection model provides an object called a *gate* to deal with this situation. A gate is a descriptor that points to a code segment and offset of a valid operating system entry point. Be-

cause the gate is a descriptor, it has a privilege level of its own, separate from that of the code segment. A program requests access to a gate by issuing a *FAR* call or jump to the gate. If no privilege violation occurs, the CPL of the executing process is set to the privilege of the new code segment, and execution continues at the privilege level of the operating system. Any parameters on the stack are copied to a new stack with the same privilege level as the operating system so that they cannot be modified by the caller. When the called routine executes a *FAR* return, the CPL is reset to that of the caller.

To prevent applications at the same privilege level from corrupting one another, the architecture provides task state segments (TSSs) and local descriptor tables (LDTs). TSSs provide direct hardware support for multitasking. When a call or interrupt is executed through a task gate, all the registers for the current process are saved in its TSS, and all the registers are reloaded with new values from the TSS pointed to by the task gate. One of the registers is the *LDTR*, which points to an LDT for the task. If the code and data segment descriptors for each task are stored in its own LDT, then there is no way that one task can get access to memory belonging to another task because it has no way to reference the other task's descriptors.

machine. By initializing the CS, DS, and SS registers with a descriptor that points to one gigantic 4-gigabyte segment, users will never have to load another segment register. A slight variation on this can permit a simple user/supervisor protection model that fits in well with the paging mechanism.

Even with all the features and enhancements found in the 386, though, there is still some room for improvement. It still has no "store pointer" instruction counterpart to the *LDS reg, [memory]* instruction for loading pointers. A set of conditional "skip" instructions would be much more efficient in compiled code than are the conditional jumps that are currently available. Compilers often have to generate the following sequence of code when processing *if* statements:

```
jnz LAB1
jmp ELSE_CLAUSE
LAB1: ; then clause
```

The jump-not-zero forces the instruction queue to be flushed, which degrades performance. Replacing the *jmp* with a *skipnz* would be more efficient.

Finally, for use in tightly coupled multiprocessor applications, Intel should have provided a hardware signal that would force the TLB to be flushed. With the current implementation, if one processor modifies a page table entry in RAM, another processor may be using the entry stored in its on-chip TLB, which could contain some invalid information.

Summary

Despite the quibbles noted above, the 80386 has been well designed. Elimination of the 64K segment size restriction makes it a pleasure to program rather than a pain. The enhanced architecture makes it easier for compilers to generate efficient code, and the machine itself is fast. The ability to run 8086 code, 286 code, and native code simultaneously means that a large software base will be available. Programmers also will appreciate the availability of new development tools. (See inset "80386 Development Tools," on page 32.)

Because of the complexity of the processor, I have been able only to highlight some of its most important new features. I hope I have spurred your interest. This chip is sure to have an important effect on microcomputing in the near future.

Bibliography

Intel Corp. *80386 Microprocessor Data Book*. 1985.
Intel Corp. *iAPX 286 Programmer's Reference Manual*. 1985.

DDJ

Vote for your favorite feature/article.
Circle Reader Service No. 3.

TNZ: An 8-bit to 16-bit Translator

by Richard A. Campbell

A serious deterrent to widespread use of a new microprocessor unit (MPU) is the lack of software available for it. This observation means that chip designers and their employers are biased toward upgrading an existing MPU rather than designing a substantially improved one. The availability of software also influences microcomputer manufacturers in their choice of an MPU, which tends to inhibit major improvements in chip design.

A major advantage of 16-bit chips when compared to 8-bit chips is simpler high-level-language compilation or fewer and more understandable lines in an assembly-language program. This leads to faster and less expensive programming because the cost of program development and maintenance is more closely related to the number of lines in a program than it is to the size of the object code or the speed of operation. In my experience, assembly-language NS320xx programs have fewer lines and instructions than their Z80 equivalents, but the object code is about the same size.

Although it is possible to write entirely new programs for these MPUs, it is often expedient to convert proven programs that have already been written. This article discusses the conversion of programs from an 8-bit MPU to a 16-bit MPU—specifically from the Z80 to the NS320xx—and includes a conversion program and some examples of converted programs.

High-Level Language

One solution to the problem of moving programs written for one MPU to

The NS320xx is not only more powerful than the Z80 but also is philosophically different.

another is to write them originally in a high-level language. Then, only a compiler, and perhaps an assembler, is necessary for the destination MPU. This has been done to a limited degree with C being used as the high-level language. The problem with this approach is that a program written in C, when compiled for an 8-bit MPU, may be unacceptably larger and slower than the same program written in assembly language. Based on their experience with 8-bit MPUs, many programmers do not believe that a compiled high-level language program can approach the size and speed of an assembly-language program.

In my experience with the NS320xx MPU, however, compiled C programs are within 50 percent of the size and speed of equivalent assembly-language programs. On a 6-MHz NS16032, for example, the compiled sieve of Eratosthenes is 149 bytes long and runs ten times in 4.2 seconds; written in assembly language, it is 141 bytes long and runs ten times in 3 seconds. This 40 percent penalty in speed for a high-level language is tolerable in many cases. In general, C-compiled programs on the NS320xx end up about 65 percent of the size of their Z80 equivalents.

Aside from the size and speed issue, the high-level-language approach is of no use for those many

useful existing programs that are available only in assembly-language form; for these some sort of conversion is necessary.

Conversion

If the new or destination MPU is merely an upgrade of, but philosophically similar to, an older one, a relatively simple conversion is possible. For example, converting an assembly-language program from an 8080 to a Z80 is a simple matter, and programs are available for doing it.

If the destination MPU is philosophically different from the original, however, the conversion becomes more difficult. For example, conversion from the Z80 to the MC6809 is harder because, even though the chips have similar power, their instruction sets are philosophically different. Converting from the 8080 or Z80 to the NS320xx is also quite difficult because the NS320xx is not only substantially more powerful than the Z80 but it is also philosophically quite different. The NS320xx instruction set is more similar to that of a VAX than to any 8-bit MPU, and assembly-language source code for the VAX is far less available than source code for 8-bit MPUs.

In order to execute a program written in Z80 assembly language on an NS320xx, you can take two conversion approaches—write a simulator program that enables the NS320xx to simulate the instructions of the Z80 or write a program that translates the Z80 program into an NS320xx program.

The problem with the simulator approach is that the simulation process will add a substantial amount to the code size and will reduce the speed of operation to a fraction of the original speed; the new MPU is figura-

Richard A. Campbell, 198 Washington Hwy., Snyder, NY 14226

Professional Print Spooling Software

COMPUTE WHILE YOU PRINT!

PrintQ®

FINALLY, A REAL MAINFRAME PRINT SPOOLER FOR YOUR IBM PC OR COMPATIBLE

If you spend part of your PC time running applications. Part of it printing. And a lot of time waiting...

Imagine a program that prints up to 10,000 pages while you run applications.

PrintQ does that. And more.

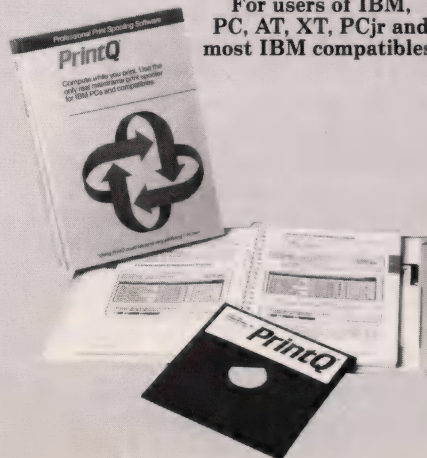
It's the first really complete print spooling subsystem for your PC. And your biggest time saver ever!

HOW **PrintQ** WORKS

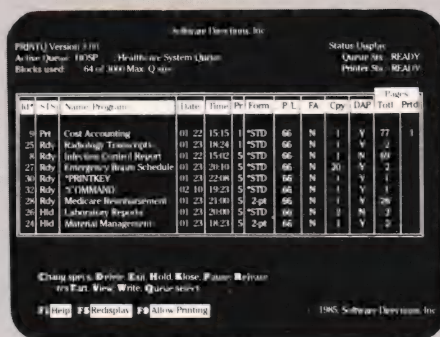
Run your application programs as usual, but instead of stopping after the print command, keep right on computing! **PrintQ** intercepts the documents bound for the printer, spools them to disk, then prints them according to your commands.

NOT COPY PROTECTED

For users of IBM, PC, AT, XT, PCjr and most IBM compatibles.



Dealer Inquiries Invited
Corporate Licensing Available
Circle no. 218 on reader service card.



"Pop up" status display lets you monitor and control document printing at any time.

WITH **PrintQ** YOU CAN...

- Re-start printing from any page.
- View documents on screen with or without printing.
- Print up to 255 copies automatically.
- Save reports for reprinting.
- Minimize form changes. **PrintQ** groups similar documents.
- Print in order of priority.
- Simplify forms alignment.
- Work while printer is down.
- Copy reports to ASCII files.
- Control from within a batch file.
- Use any printer (including laser).
- No program modifications required.

... AND MUCH MORE.

"Using **PrintQ** could become very addicting, and users may never want to go back to ordinary printing again."

— PC Week

FULLY GUARANTEED

Use **PrintQ**. If you're not convinced **PrintQ** saves time, increases productivity and enhances printer function, return it within 30 days for a full refund.

Order now and you'll receive the **PrintQ** disk, complete easy to understand documentation, and the Quick Start Card which will have you reaping the benefits of **PrintQ** in seconds.

Call toll-free or mail the coupon to order now.

1-800-346-7638

In New Jersey Call
201-584-8466
Same day shipping on phone orders

SDI Software Directions, Inc. 1572 Sussex Turnpike, Randolph, NJ 07869

YES. Rush me **PrintQ** for just \$89, which includes postage and handling (Canada — add \$10; Foreign — add \$20). If I'm not convinced **PrintQ** saves time, increases productivity and enhances printer function, I'll return it within 30 days for a full refund.

Name

Company

Address

City State Zip

☐ Check enclosed. ☐ Visa ☐ MasterCard ☐ AmEx.

Acct. No. Exp. date

Signature Phone #

NJ residents add 6% sales tax.

IBM is a registered trade mark of International Business Machines Corporation.

tively running with both hands tied behind its back. Thus, if you are interested in having a program that can be executed with the additional speed and power of the NS320xx, the translation approach makes more sense. Indeed, why should you bother to convert a program if it will not execute any better than it did on the original MPU?

An additional factor to consider is whether or not the program you want to convert is arithmetic, partic-

ularly floating-point arithmetic, intensive. If it is, then the translation approach will lead to a markedly better (smaller, faster, and more powerful) program than the simulation approach. This is because the larger operands and the floating-point coprocessor permit floating-point-number input, arithmetic, and output to be done with far fewer instructions than are necessary with an 8-bit chip. For example, the code for a BASIC interpreter with floating-point arithmetic translated to the NS320xx is roughly one-half the size of the original Z80 version.

SETVAL:	CALL	TSTV	;IS A VARIABLE NAME?
	JR	Z,QWHAT	;“WHAT?” NO VARIABLE
	PUSH	HL	;SAVE ADDRESS OF VAR.
	LD	A,'='	
	CALL	TSTC	;IS “=” SIGN
	JP	NZ,QWHAT	;NO ==??
	CALL	EXPR	;EVALUATE EXPR.
	LD	B,H	;VALUE IN BC NOW
	LD	C,L	
	POP	HL	;GET ADDRESS
	LD	(HL),C	;SAVE VALUE
	INC	HL	
	LD	(HL),B	
	RET		;ADDRESS + 1 OF VAR IN R3

Table 1: Original Z80 LET statement interpreter

SETVAL:	CALL	TSTV	;IS A VARIABLE NAME?
	BEQ	QWHAT	;“WHAT?” NO VARIABLE
	MOV.D	R3,TOS	;SAVE ADDRESS OF VAR.
	MOV.B	'=',R0	
	BSR	TSTC	;IS “=” SIGN
	BNE	QWHAT	;NO ==??
	BSR	EXPR	;EVALUATE EXPR.
	MOV.B	RH,RB	;VALUE IN BC NOW
	MOV.B	R3,R1	
	MOV.D	TOS,R3	;GET ADDRESS
	MOV.B	R1,(R3)	;SAVE VALUE
	ADDQ.W	+1,R3	
	MOV.B	RB,(R3)	
	RET		;ADDRESS + 1 OF VAR IN R3

Table 2: Original TNZ output for LET interpreter

SETVAL:	CALL	TSTV	;IS A VARIABLE NAME?
	BEQ	QWHAT	;“WHAT?” NO VARIABLE
	MOV.D	R3,TOS	;SAVE ADDRESS OF VAR.
	MOV.B	'=',R0	
	BSR	TSTC	;IS “=” SIGN
	BNE	QWHAT	;NO ==??
	BSR	EXPR	;EVALUATE EXPR, VALUE IN R3
	MOV.D	TOS,R1	;GET ADDRESS BACK
	MOV.D	R3,0(R1)	;SAVE VALUE IN R3
	RET		;ADDRESS OF VAR IN R1

Table 3: Improved LET interpreter

You must also decide how much “automatic” translation should be done and how much if any “hand” translation will be required. I developed the translation program described here to translate three different Z80 programs, and initially I envisioned a completely automatic translation. One of the programs—the BASIC interpreter mentioned earlier—had a substantial amount of floating-point arithmetic. As the translation program developed, however, it became apparent that a totally automatic translation approach was forcing the NS320xx to simulate the Z80; the translated code was three times the size of the original and would have executed more slowly than the original. Thus, it became evident that a significant amount of hand editing was necessary in order to end up with a really useful NS320xx program. Some of the reasons for this will become apparent when you consider the two chips and the translation program in more detail.

This means that this translation program, TNZ, is not automatic. It can do most of the conversion process but is not intended to and generally will not yield a fully executable program. Some editing is necessary to get a program that will run; insightful editing will yield a very efficient program.

The NS320xx

Briefly, the NS320xx has eight general-purpose registers that are 32 bits wide. It can carry out 8-, 16-, or 32-bit arithmetic and logical operations with operands in the registers that are addressed through registers or in memory. An operand can be either data or an address. The direction of movement in an instruction is from left to right, opposite to that of a Z80 instruction. The size of the operand is specified in the instruction—for example, *.B* = byte, *.W* = word (2 bytes), and *.D* = double-word (4 bytes).

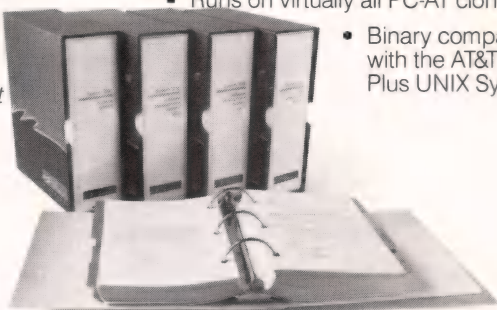
The floating-point coprocessor (FPU) can carry out either 32-bit *F* single-precision or 64-bit *L* double-precision operations on operands either in the FPU's registers or in memory, as well as integer to floating-point to integer conversions. It does not do well with operands in an MPU register.

All arithmetic and logical instructions can be performed on operands

If you think you can't afford a UNIX® system. we've got a \$160 surprise.

Turn your PC into a multi-user system.

Convert your IBM PC-AT (or compatible) into a multi-user/tasking UNIX work station—at *absolutely the best price* anywhere, any time. Based on the AT&T-certified UNIX System V/286, the MICROPORT SYSTEM V/AT is designed for use in virtually any computer environment, from office automation to software development.



- Dynamic disk buffer allocation provides RAM disk performance for systems with large memory configuration.
- Runs on virtually all PC-AT clones.
- Binary compatible with the AT&T 6300 Plus UNIX System.

Over 200 utilities come standard.

Grep, awk, sort, split, cut, paste, vi and ed (and many more) now let you search and modify files, make use of electronic mail, emulate terminals, calculate electronically, convert data and publish.

SYSTEM V/AT is more than a look-alike. It was derived from AT&T's own UNIX System V release 2 iAPX286. It thereby contains standard System V features the competitors don't support, such as the powerful symbolic debugger, sdb, the shell-layering job-control facility and the F77 Fortran compiler, as well as programming tools such as ctrace, cflow, and bs. Also standard is File System Hardening which greatly reduces data loss in a power failure.

Want some more features?

- Console driver providing ANSI terminal interface for monochrome, CGA, Hercules and EGA cards.
- Multiple Virtual consoles allow up to four virtual windows of operation.
- Record and File Locking
- Supports the 286's 16 megabyte virtual address space and fully utilizes its other advance features.
- Supports all standard IBM drive types and most non-standard hard-disk drives.
- Requires only one hard-disk partition, and allows DOS to reside on the same hard disk.
- Provides utilities to transfer files to-and-from DOS file systems.

Super software- development environment

We've provided everything: Make, yacc, lex, sccs, cflow, ctrace plus every standard System V software-development tool. The F77 Fortran compiler. And the AT&T Portable C compiler for the 286. Both C and Fortran compilers generate 287 instructions directly—for systems not containing 287 math coprocessors, a kernel-resident IEEE-compatible 287 emulator is provided. The large-model code produced by the compiler is among the densest and fastest currently available.

So, how do we do it?

MICROPORT offers SYSTEM V/AT at a fraction of the price of the competitors simply because we build on the generic System V/286 product from AT&T. This entire utility package from the certified release has been copied directly to SYSTEM V/AT—without so much as a recompile. Not only does this mean that MICROPORT can offer SYSTEM V/AT at a remarkable low price, it also *guarantees a level of quality* present in few (if any) other UNIX-system implementations. (And, since our staff was part of the group that implemented the standard System V/286 port for Intel, MICROPORT can offer comprehensive support for the system, as well.)

And a dollar change

The price is even better than you thought. Order right away and we'll return one silver dollar just as rapidly, with your product shipment. (If you'd like a little more time we'll apply that dollar to the cost of a brochure—which we'll send right away too.)

NOTE: Educational institutions—Call about our nominally-priced site licensing and source code

60 DAY MONEY-BACK GUARANTEE

And a dollar change.



To order: Complete the information below. Your attractively-packaged and fully-documented order will be shipped within two weeks.

MICROPORT SYSTEMS, INC.

4200 Scotts Valley Drive
Scotts Valley, CA 95066
408/438-UNIX or 800/PC2-UNIX (outside CA)

SYSTEM V/AT

- ☐ **RUNTIME SYSTEM** Includes the SYSTEM V/AT operation system and over 200 utilities. for two users.
QUANT: _____ **\$160.00**
- ☐ **SOFTWARE DEVELOPMENT SYSTEM** The complete Software Generation System for 286 development.
QUANT: _____ **\$169.00**
- ☐ **TEXT PREPARATION SYSTEM** Includes nroff, troff, spell and other programs.
QUANT: _____ **\$169.00**
- ☐ **THE COMPLETE SYSTEM** Contains all three packages indicated above.
QUANT: _____ **\$439.00**
- ☐ **OPTIONAL** three to eight-user upgrade.
QUANT: _____ **\$99.00**

Subtotal: _____

(CA residents add 6.5% tax per copy) _____

Shipping and handling charges (In the USA, \$14.00; in Canada, \$18.00; and in Europe, \$110.00) _____

TOTAL DUE: _____

NAME _____

TELEPHONE _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

COUNTRY _____

☐ VISA ☐ MASTERCARD ☐ BANK DRAFT ☐ CHECK

CARD NUMBER _____ EXP DATE _____

☐ Send a brochure only and keep me on your mailing list, please.


M I C R O P O R T

UNIX and DWB are trademarks of AT&T.
IBM and IBM PC-AT are trademarks of IBM CORPORATION.
SYSTEM V/AT is a trademark of MICROPORT SYSTEMS, INC.

Circle no. 154 on reader service card.

in memory addressed in several different modes (including the stack pointer) so a program may use no registers at all! Indeed, unless an operand is accessed repeatedly, it is just as well left in memory.

The Translation Program

The program itself is based on a generic assembler program that I have used for various MPUs, including the 8080, MC68000, and NS320xx. The source is about 1,200 lines long, and the 8080 code is 18,000 bytes (NS320xx code is 11,500 bytes), about the same size as the 8080 assembler but much shorter than the other assemblers. Remnants of the other programs remain, for which I apologize.

TNZ operates like an assembler does in that its input is an assembly-language file. It differs in that it uses only a single pass and its output is another assembly-language file—in ASCII—instead of the bit patterns produced by an assembler. By use of *#ifdefs* it can be compiled by two C compilers that differ with regard to their disk input/output library functions.

It consists of four files: TNZH.C, TNZ.C, TNZ2.C, and TNZO.C. The first is the header file, which contains constants and global variables. The second is the main body of the translation process. The third consists of fairly routine functions for initialization, opcode-table loading and lookup, integer conversion, file input/output, and so on. The fourth consists of the opcode table, which is read

from the disk during the initialization process. The opcode entry for each operation and register has the source (Z80) opcode in ASCII; the destination (NS320xx) opcode in ASCII; and two 16-bit words of operation class, size, and modifier.

In TNZ, the *main()* function sizes memory, calls the initialization function (which reads in the opcode table, checks for user options, and opens the output file), and then calls *doscan()*. The program is capable of translating multiple files, so *doscan()* opens the input file(s) and starts reading, converting, and outputting each line of the input.

The heart of *doscan()* is a loop through *rdline()*, *doline()*, and *outcode()*. Each noncomment line is parsed in *rdline()*: symbols are simply copied to the output in *rdsymbol()*; operations and modifiers are converted in *rdopcode()*; and operands are converted in *rdoperands()*. At this point, the converted operation and operands are in temporary buffers.

Next, in *doline()*, one of three actions can occur, depending on the particular operation being translated. One, in the case of a simple translation, the temporary buffers are copied, with operation size added and operands reversed, to the output buffer in *doarmov()*. Two, special conversion steps are carried out in *dospes()*—for example, Z80 operations that have an implied accumulator destination need to have the destination added explicitly. Three, program flow operations are handled in *doprfl()*—for example, Z80 conditional calls and returns need special handling.

In any event, after *doline()* the converted line is ready to be passed to the output file by *outcode()*. Then, the next line is read in *rdline()* and the above loop is repeated until the end of the file, at which point *doscan()* closes the input file and either moves to the next input file or closes the output file and stops.

This is a general summary of the program. Most of it is straightforward programming; the tricky stuff is in *dospes()* and *doprfl()* when a simple translation is not possible. Because I use it with two different C compilers, I have tried to avoid compiler-specific programming.

```
;TEST IF TEXT IS A NUMBER,IF NOT RET 0 IN B & HL
TSTNUM:  LD      HL,0
         LD      B,H
         CALL    IGNBLK      ;NEXT NON-BLANK CHAR
TSTNML:  CP      '0'
         RET     C
;IF NUMB, MAKE BINARY IN HL & A=NO. OF DIG.
         CP      ':'
         RET     NC
         LD      A,0FH
         AND     H
         JP      NZ,QHOW
         INC     B
         PUSH    BC
         LD      B,H      ;HL=10;HL+(NEW DIGIT)
         LD      C,L
         ADD     HL,HL
         ADD     HL,HL
         ADD     HL,BC
         ADD     HL,HL
         LD      A,(DE)
         INC     DE
         AND     0FH
         ADD     L
         LD      L,A
         LD      A,0
         ADC     H
         LD      H,A
         POP     BC
         LD      A,(DE)
         JP      P,TSTNML
```

Table 4: Z80 string-to-number conversion

Translation Problem Areas Half Registers

Direct translation of many Z80 register operations involving 16-bit and many 8-bit operands presents no problem. However, you can access directly only the lower 8 or 16 bits in the NS320xx registers. This means that the practice of directly accessing the upper 8-bit halves of the Z80—BC 'B', DE 'D', HL 'H' (and IX and IY registers, if you're cute) is not possible on the NS320xx.

You can solve this problem by simulating Z80 registers in memory and accessing their two bytes separately, but this is not a very efficient solution. What I did was to treat the NS320xx registers arbitrarily as follows—R0 = AF, R1 = C or BC, R2 = DE or E, and R3 = HL or L and to use simulated registers in memory for the high halves B, D, and H.

If the high and low half registers are being used as separate operands, this works well. If they are actually the two bytes of one 16-bit operand, though, then some hand changes are necessary.

Say you wanted to load DE with the data at (HL):

```
LD      E,(HL)
INC     HL
LD      D,(HL)
```

TNZ will translate this to:

```
MOV.B   (R3),R2
ADDQ.W  +1,R3
MOV.B   (R3),RD
```

Although this translation will not work properly, you can change it to the one instruction MOV.W (R3),R2—the ADDQ.W +1,R3 is probably unnecessary. This change is quite easy to spot by eye, but for TNZ to figure this out would take a more insightful translator program than I am willing to write.

Flags

A great difference exists between the two MPUs' flags and their use. Again, you could simulate the Z80 flag actions, but this would result in a great deal of often unnecessary code and we are interested in an efficient translated program. The Z80 has three constantly used flags—zero, carry, and positive. The same flags are affected

by arithmetic, logical, and comparison operations. The NS320xx has a similar set of flags, but they operate differently. Arithmetic operations affect the carry flag only. Logical opera-

tions—AND, OR, and so on—do not affect the flags. Comparisons affect the equal (zero), less-than, greater-than (signed), lower-than, and more-than (unsigned) flags. Thus, an explicit

```
;Z80 - NS320XX Translator; RAC 03/26/86
;TEST IF TEXT IS A NUMBER,IF NOT RET 0 IN B & HL
TSTNUM:  MOV.W      0,R3
          MOV.B      RH,RB
          BSR        IGNBLK      ;NEXT NON-BLANK CHAR
TSTNML:  CMP.B      '0',R0
          BLS        XYZ1
          RET
XYZ1:    ;IF NUMB, MAKE BINARY IN HL & A=NO. OF DIG.
          CMP.B      ' ',R0
          BHI        XYZ2
          RET        ;???
XYZ2:    MOV.B      0F0,R0
          AND.B      RH,R0
          CMPQ.B     0,R0
          BNE        QHOW
          ADDQ.B     +1,RB
          MOV.D      R1,TOS
          MOV.B      RH,RB      ;HL=10;HL+(NEW DIGIT)
          MOV.B      R3,R1
          ADD.W      R3,R3
          ADD.W      R3,R3
          ADD.W      R1,R3
          ADD.W      R3,R3
          MOV.B      (R2),R0
          ADDQ.W     +1,R2
          AND.B      0FH,R0
          ADD.B      R3,R0
          MOV.B      R0,R3
          MOV.B      0,R0
          ADDC.B     RH,R0
          MOV.B      R0,RH
          MOV.D      TOS,R1
          MOV.B      (R2),R0
          CMPQ.B     0,R0
          BLE        TSTNML      ;???
;
```

Table 5: TNZ output for string-to-number convertor

```
;TEST IF TEXT IS A NUMBER; RETURN NO. IN R3, DIGITS IN R2
;R5 CONTAINS TEXT POINTER.
;IGNBLK RETURNS NEXT NON-BLANK CHARACTER IN R0
;
TSTNUM:  MOVQ.D      0,R3      ;ZERO NUMBER
          MOVQ.B      0,R2      ;ZERO DIGITS
          BSR        IGNBLK      ;NEXT NON-BLANK CHAR
TSTNML:  AND.D      7FH,R0      ;STRIP HIGH BITS
          SUB.B      '0',R0
          BCS        TSTNMZ      ;IS NG, RETURN
          CMP.B      10,R0      ;< 10?
          BHI        TSTNM2      ;OK, TAKE CHAR
TSTNMZ:  RET
TSTNM2:  ADDQ.B      1,R2      ;COUNT THE DIGIT
          MUL.D      10,R3      ;MULTIPLY NO. BY 10
          ADD.D      R0,R3      ;ADD IN NEW DIGIT
          ADDQ.D      1,R5      ;MOVE TO NEXT CHAR
          MOV.B      0(R5),R0    ;GET NEXT CHAR
          BR         TSTNML      ;GO FOR NEXT
;
```

Table 6: Improved string-to-number convertor

TNZ

(continued from page 45)

comparison is required to set the equal flag, for example.

TNZ keeps track of previous instructions, and if a conditional jump, call, or return is encountered that has not been preceded by an explicit comparison, then the code for a comparison with zero is produced to set the flags. This is less than perfect, particularly when a subroutine has been called that is to return informa-

tion in the flags. There is no way for the translator to know what has been done in the subroutine, so a comparison with zero may not be appropriate and is not carried out.

Examples

The following is an example of some original translated and edited code that is not particularly arithmetic in nature. This is the *(LET) X = ??* part of an integer BASIC interpreter. The code ends up about the same size as the original, but double-precision val-

ues are being handled. To handle floating-point values would involve trivial changes on the NS320xx but would involve substantial modification to the Z80 program.

Table 1, page 42, shows the Z80 code, and Table 2, page 42, shows this code after TNZ has translated it. Notice the half-register problems involving *RB*. These two instructions and the *ADDQW +1,R3* need to be eliminated, and the size of the *R3,R1* and *R1,R3* moves are changed to double words—*D*—a relatively simple editing task. When double-precision integer arithmetic is executed, the translated code becomes that shown in Table 3, page 42.

Another example of translated code involves the conversion of a text string into a binary number. This somewhat arithmetic example points out the advantages of the translation rather than simulation approach because a great many of the instructions needed for the Z80 can be eliminated. Note that this routine as finally translated can be used for floating-point number conversion with only a few changes.

Table 4, page 44, shows the Z80 routine, and Table 5, page 45, shows the translated version. Notice that *RET C* has been changed into the sequence *BLS XYZ1, RET, XYZ1:...* because the preceding operation was a comparison and because the NS320xx has no conditional-return instructions. This routine, as directly translated, is loaded with half-register problems and would not work. Some editing, however, will result in the routine shown in Table 6, page 45, which is much shorter and more powerful because it handles 32-bit integers of nine or more digits. Note in the test following the *SUBB '0',R0* instruction in Table 6 that the carry flag is tested, and following the *CMPB 10,R0*, the higher-than flag is tested.

DDJ

Vote for your favorite feature/article.
Circle Reader Service No. 4.

The listing for this article is presented in a machine-readable form—a Softstrip produced by Cauzin Systems. The strips begin on page 84. The text of the listing is available for downloading in the DDJ Electronic Edition on CompuServe.

Demo Disk
Available



MACH 2 TURBOCHARGES BASIC

Turbo Pascal Version
coming soon!

Brand new from the publishers of The Inside Track and Peeks 'n Pokes, MACH 2 is a library of ultra fast assembler subroutines for interpreted and compiled BASIC. Many functions faster than Turbo Pascal and C. Debug interpreted and compile the same program with no changes.

MACH 2 includes: Window Manager (also draws boxes) • Extra Memory Manager—store/sort/search data using all available DOS memory • Controlled input routine ignores Ctrl-C and Ctrl-Break • Display data 4-10 times faster than BASIC—instantly when compiled • BLOAD/BSAVE in compiled BASIC • Read/write files at DOS speeds • Scroll windows any direction • Print using for numbers up to 6 times faster than BASIC • Change file attributes—hide, unhide, read-only, etc • Get & change default drive/current directory • DOS/BIOS function calls and interrupts • Many more functions, sample programs and a manual explaining the use of each routine.

No assembler program or knowledge of assembler programming is required. MACH 2 can be used by beginners.

Still not convinced? Send us a check for \$5 and you will receive a demo disk. That \$5 will be deducted from the purchase price of MACH 2.

MACH 2 is only \$75.00 and is not copy protected. Include our subroutines in your programs with no royalties. Requires DOS 2.00 +, GW-BASIC or IBM BASIC. Video routines require IBM or compatible. Others run under any MS-DOS 2.00 +. Compatible with Quick BASIC 1 & 2, BASCOM 1 & 2, MS compilers.

SHIPPING AND HANDLING. \$3 USA, \$5 1st class (Canada) \$18.00 elsewhere. We welcome MC/Visa/COD (add \$3 COD). To order call 1-800-922-3383. In Georgia, or for tech support, call 404-973-9272.

MicroHelp, Inc. • 2220 Carlyle Drive • Marietta, Georgia 30062

Circle no. 215 on reader service card.

SAS Institute Inc. Announces

Lattice C Compilers for Your IBM Mainframe

Two years ago...

SAS Institute launched an effort to develop a subset of the SAS® Software System for the IBM Personal Computer. After careful study, we agreed that C was the programming language of choice. And that the Lattice® C compiler offered the quality, speed, and efficiency we needed.

One year ago...

Development had progressed so well that we expanded our efforts to include the entire SAS System on a PC, written in C. And to insure that the language, syntax, and commands would be identical across all operating systems, we decided that all future versions of the SAS System—regardless of hardware—would be derived from the same source code written in C. That meant that we needed a C compiler for IBM 370 mainframes. And it had to be good, since all our software products would depend on it.

So we approached Lattice, Inc. and asked if we could implement a version of the Lattice C compiler for IBM mainframes. With Lattice, Inc.'s agreement, development began and progressed rapidly.

Today...

Our efforts are complete—we have a first-rate IBM 370 C compiler. And we are pleased to offer this development tool to you. Now you can write in a single language that is source code compatible with your IBM mainframe and your IBM PC. We have faithfully implemented not only the language, but also the supporting library and environment.

Features of the Lattice C compiler for the 370 include:

- **Generation of reentrant object code.** Reentrancy allows many users to share the same code. Reentrancy is not an easy feature to achieve on the 370, especially if you use non-constant external variables, but we did it.
- **Optimization of the generated code.** We know the 370 instruction set and the various 370 operating environments. We have over 100 staff years of assembler language systems experience on our development team.
- **Generated code executable in both 24-bit and 31-bit addressing modes.** You can run compiled programs above the 16 megabyte line in MVS/XA.
- **Generated code identical for OS and CMS operating systems.** You can move modules between MVS and CMS without even recompiling.
- **Complete libraries.** We have implemented all the library routines described by Kernighan and Ritchie (the informal C standard), and all the library

routines supported by Lattice (except operating system dependent routines), plus extensions for dealing with 370 operating environments directly.

Especially significant is our byte-addressable Unix®-style I/O access method.

- **Built-in functions.** Many of the traditional string handling functions are available as built-in functions, generating in-line machine code rather than function calls. Your call to move a string can result in just one MVC instruction rather than a function call and a loop.

In addition to mainframe software development, you can also use our new cross-compiler to develop PC software on your IBM mainframe. With our cross-compiler, you can compile Lattice C programs on your mainframe and generate object code ready to download to your PC.

With the cross-compiler, we also offer PLINK86™ and PLIB86™ by Phoenix Software Associates Ltd. The Phoenix link-editor and library management facility can bind several compiled programs on the mainframe and download immediately executable modules to your PC.

Tomorrow...

We believe that the C language offers the SAS System the path to true portability and maintainability. And we believe that other companies will make similar strategic decisions about C. Already, C is taught in most college computer science curriculums, and is replacing older languages in many. And almost every computer introduced to the market now has a C compiler.

C, the language of choice...

C supports structured programming with superior control features for conditionals, iteration, and case selection. C is good for data structures, with its elegant implementation of structures and pointers. C is conducive to portable coding. It is simple to adjust for the size differences of data elements on different machines.

Continuous support...

At SAS Institute, we support all our products. You license them annually; we support them continuously. You get updates at no additional charge. We have a continuing commitment to make our compiler better and better. We have the ultimate incentive—all our software products depend on it.

For more information...

Complete and mail the coupon today. Because we've got the development tool for your tomorrow.



SAS Institute Inc.
SAS Circle, Box 8000
Cary, NC 27511-8000
Telephone (919) 467-8000 x 7000

I want to learn more about:

- ☐ the C compiler for MVS software developers
- ☐ the C compiler for CMS software developers
- ☐ the cross-compiler with PLINK86 and PLIB86

today...so I'll be ready for tomorrow.

Please complete or attach your business card.

Name _____
Title _____
Company _____
Address _____
City _____ State _____ ZIP _____
Telephone _____

Mail to: SAS Institute Inc., Attn: CC, SAS Circle, Box 8000, Cary, NC, USA.
27511-8000. Telephone (919) 467-8000, x 7000

DDJ 10/86

Modula-2 Compilers for the IBM PC

by Namir Clement
Shammas

*Four compilers from
the latest generation
are compared.*

This comparative review looks at four Modula-2 compilers for the IBM PC. The compilers come from Logitech, Interface Technology Corp. (ITC), Modula Corp., and PColliers Systems.

Logitech offers its Modula-2/86 overlayed four-pass compiler and linker in three configurations: a base system, a base system with 8087 support, and Modula-2/86 Plus. For this review I evaluated Modula-2/86 Plus, which I'll refer to as simply Modula-2/86 for the rest of the review. This is a full system that runs on machines with 512K RAM, taking advantage of the large memory to increase the speed of compilation. Support for the 8086, 8087, 80186, 80286, and 80287 chips is included, and it comes with a fully linked compiler and linker. Logitech also includes the MOD text editor, which can become the heart of an integrated software-development system.

ITC sells the Modula-2 System Development System (M2SDS), a window-based integrated system containing a compiler, linker, librarian, and built-in syntax-oriented editor. The company also sells an advanced software-development version, SDS-XP, which includes an extended library, the M2MAKE utility, and the Foreign Object Module Importer (FOMI). The FOMI utility enables software developers to import object modules created by an 8088/8086 assembler into the SDS library. I received the M2SDS version for this review.

Modula Corp. has launched its new PC Modula-2, a one-pass native compiler, to replace a previous version

that generated much slower code. PC Modula-2 has neither a built-in editor nor an integrated environment.

PColliers Systems, a newcomer to the arena of Modula-2 vendors, offers Modula-2PC, a one-pass native compiler. The company will come out with 8087 support, an editor, and a debugger in late 1986.

Compilers and Linkers Modula-2/86

You can compile one or more source code files with the four-pass Modula-2/86 compiler in one of two ways. The source file names can follow *M2 COMP*, which you type at the DOS command level. This is interpreted by the Modula-2 shell as a request to compile the listed files and then exit back to DOS. In the second method, you simply type *M2 COMP* from DOS to invoke the compiler, which in turn prompts you for a source file name. At the end of compilation, another similar prompt appears. This allows you to compile another file or exit to DOS by pressing the Escape key. Logitech also provides a fully linked compiler (M2C.EXE), which runs faster than the standard overlayed version. The Modula-2/86 compiler produces LOD files that run under the Modula-2 shell. The LOD2EXE program is used to convert the LOD files into stand-alone .EXE files that

run directly from DOS.

The compiler has several directives or switches, whose default states are shown in the manual. The query and auto-query switches alter the file-search mechanism and strategy. The default search uses the module name to construct the file name. There is a switch for the systematic generation of a listing and another switch that invokes the latter only when the compiler detects an error. The listings can include a header, a footer, and a date—each included by turning on a switch. There's an emulator/coprocessor switch to control the generation of in-line 8087 code as well as a switch to create code for the 80186/80286. The default is, of course, 8086/8088 code. Other switches are used to perform stack test, range, and overflow checking and index and nil pointer tests. The alignment option positions all multi-byte variables on even address boundaries. The collection of switches can be included inside comments in the source files.

You invoke the Modula-2/86 linker in a fashion similar to the way you invoke the compiler. Again, linked file names can be included on the DOS command line. Moreover, there is a fully linked version (M2L.EXE). The inclusion of file names on the DOS command line makes possible the use of batch files to invoke the compiler and linker. Using the fully linked versions of the compiler and linker, the whole process becomes even faster.

The linker has several options, including a base-layer option for use in compiling overlays. The query and auto-query options operate similarly to those of the compiler. The large option enables the number of linked

Namir Clement Shammas, 4814 Mill
Park Ct., Glen Allen, VA 23060

How to tackle a 300 page monster.

Turn your PC into a typesetter.

If you're writing a long, serious document on your IBM PC, you want it to look professional. Precise. Easy to read. You want MicroT_EX.

MicroT_EX was designed especially for desktop publishers who require heavy duty typesetting. It is based on the T_EX standard, with tens of thousands of users worldwide. Documents from smaller than 30 pages to 5000 pages or more. And that's something that other programs just can't match.

No other PC software gives you as many advanced capabilities as MicroT_EX. Superior hyphenation control, the sophistication of ligatures (ffi, fi) and kerning; down-loadable fonts; aesthetic handling of math ($\pi = f'(x)$), and foreign language characters; complex table construction and multi-column tasks; dot matrix, laser printer and phototypesetter output. When used with our L^AT_EX macro package, it automatically enumerates and cross-references pages, sections, footnotes and illustrations. Plus it automatically creates your indexes, tables of contents, and even updates them for you after last minute insertions.

So if you want typesetting software that's as serious as you are about your writing, get MicroT_EX. **Call toll free 800-255-2550** or 617-944-6795 to order or for more information.* Order with a 60-day money back guarantee.

Now available for Macintosh.

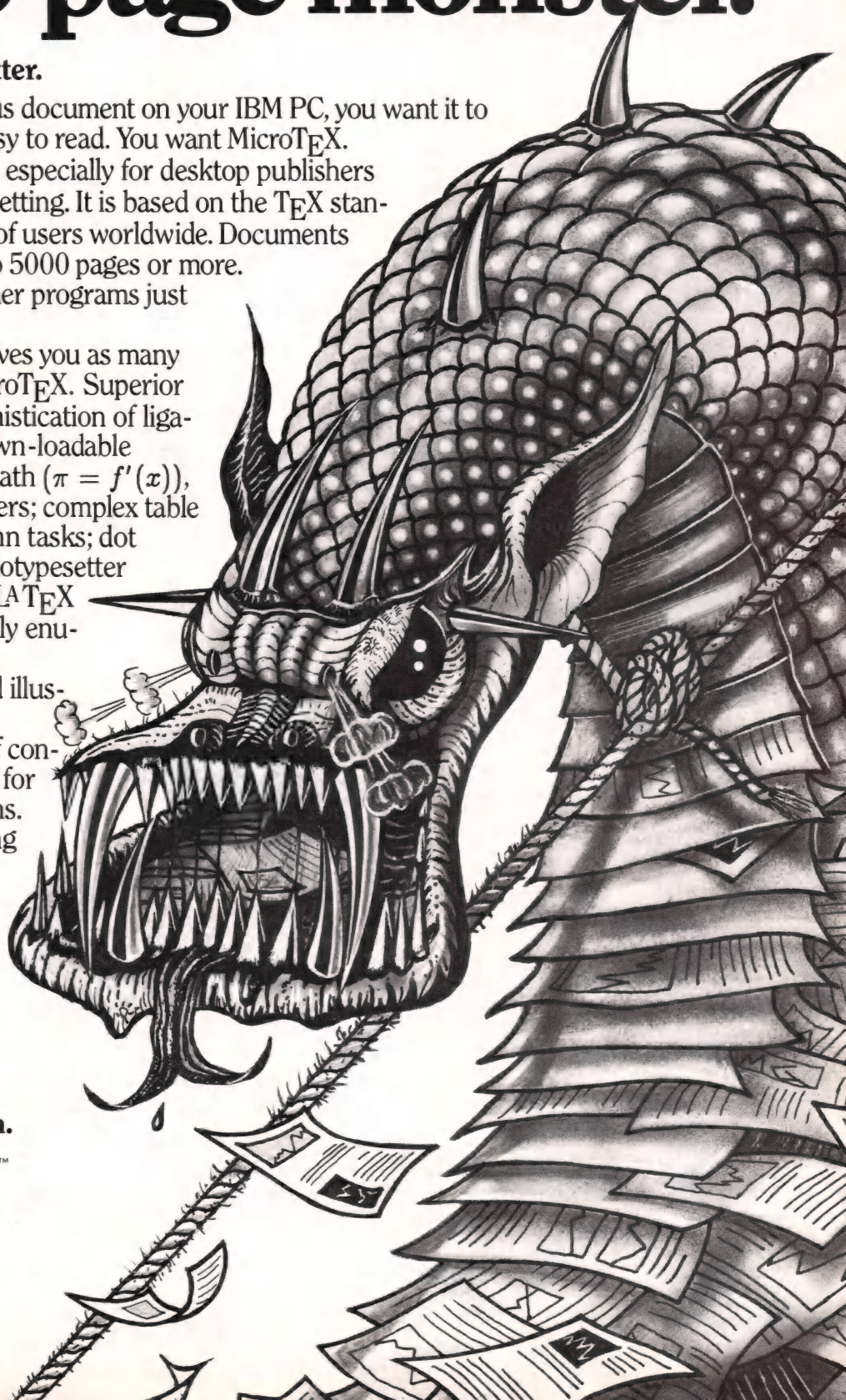
MicroT_EX[™]

from Addison-Wesley

Serious typesetting for serious desktop publishers.

*Dealers, call our Dealer Hot Line: 800-447-2226 (In MA, 800-446-3399), ext. 2643.

Circle no. 92 on reader service card.



Some developers have Apple in their hip pocket.

And so can you.

Simply by joining APDA—the Apple Programmer's and Developer's Association.

APDA is the one-stop source for the Apple programmer or developer. Members will enjoy timely and easy access to the most current tools of the trade.

Like Apple® II and Macintosh™ development tools, system software, utilities and documentation from Apple Computer. Language products from leading software manufacturers. A wide selection of technical books, including Addison-Wesley's entire Apple Technical Library. Even technical notes and

Apple manuals.

You can have a whole world of Apple data at your fingertips.

And, as our membership grows, you can expect more services to meet your expanded needs.

APDA was created to serve the entire Apple programming community, and is open to anyone who needs advanced tools and information in order to create Apple-compatible products:

Educators and students. Hackers and hobbyists. Consultants and VARs. And professional hardware and software developers of every stripe.



Apple Programmer's & Developer's Association

290 SW 43rd Street • Renton, WA • 206-251-6548

Signature Bill Budge

APDA was founded by APPL.E.CO-OP, one of the first and largest Apple users' groups. We've got the full cooperation of Apple Computer, Inc.

For just \$20 per year, we can give you all of the above. Plus a data-packed quarterly catalog/newsletter, update bulletins and access to our on-line ordering system.

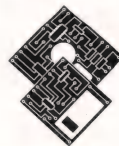
If you join APDA before 1987, you'll receive your choice of a free Apple II or Macintosh technical book worth more than the price of admission.

So call now.
206-251-6548.

Or write: Apple Programmer's and Developer's Association, 290 SW 43rd St., Renton, WA 98055.

We can't guarantee you'll write the same great code as *some* of our members.

But it won't be because of lack of information.



Apple Programmer's & Developer's Association

Apple and the Apple logo are registered trademarks of Apple Computer, Inc. Macintosh is a trademark of McIntosh Laboratories and is being used with its express permission.

Circle no. 231 on reader service card.

C spoken here...

High C™

Do you want to use a C compiler that

- was chosen by Ashton-Tate for implementing dBASE III® Plus
- was well rated in *Computer Language*, Feb. 86 and *Dr. Dobbs Journal*, August 86.
- "would have saved me three weeks of porting time had I had High C instead of Microsoft's new C"
- Mike LeBlanc, compiler developer, Sky Computers
- "is the only C compiler for the IBM PC capable of compiling NYU's Ada/Ed compiler"

Dave Shields, research scientist, New York Univ.

- has a complete run-time library
- has structure assignment, **enum**, **void**...
- supports nested functions as in Pascal
- supports pcc and full K&R C plus some latest, nifty extensions from the new ANSI-proposed C standard
- "saved 15% of code over five large modules of MultiMate relative to Lattice C"

David Beauchesne, Multimate International

Pascal spoken here...

Professional Pascal™

Do you want to use a Pascal compiler that

- was chosen by Lifetree Software, Inc., for implementing Volkswriter Deluxe™
- was well rated in *Computer Language*, May 86: "The clear choice for large-scale programming projects..."; and *PC Tech Journal*, July 86
- serves as a systems and applications language at CAD/CAM giant Daisy Systems Corporation
- has 8-, 16-, and 32-bit integers; sets up to 64K bits
- has varying-strings of up to 64K characters
- has a full-fledged C macro preprocessor
- has many run-time library additions: UNIX™-like I/O, multiple heaps, interrupts, . . .
- has all the bit-pushing operators of C
- has many more extensions, getting you half way to Ada® for a non-Ada price
- is the "howitzer" of Pascals and "could well be the most powerful Pascal compiler ever implemented on a microcomputer" *PC Magazine*, Oct. 29, 1985, p. 144



Power Tools

Power Users

for

Each compiler • generates **superb code**, with optimizations such as common-subexpression elimination and cross-jumping • sports no less than **five memory models** for the 8086 (Small, Compact, Medium, Big, and Large) • supports a unique implementation of register variables • supports the **8087/80287** in native mode, or **emulates** • supplies **three** floating-point formats • generates special instructions for the 80186/286 • generates code that runs in **80286 protected mode** • gives you **hundreds of error and warning messages**, helping you find those subtle bugs *before* you need a debugger's help • lets you **overlay data** as well as code (when used with PLINK86), for substantial space savings • lets you write **interrupt routines** directly in high-level language • lets you get "close to the machine" with built-in move/scan/compare operations • is supported by equivalent **resident and cross** compilers for the 80286 (UNIX V.2, Xenix, Concurrent DOS 286), 80386 (DOS, UNIX V.3), 68010/20/68881 (UNIX V & 4.2, GEM-DOS™), 32032 (UNIX 4.2), VAX (UNIX 4.2, VMS), IBM 370, . . . • contains a **multi-modular cross-referencer** • produces ROM-able code for embedded applications • can talk to those **other languages** by those other vendors • gives you **64K run-time stack** space that can be shared with the heap • is endowed with an **amazing number of pragmas** (compiler controls) for customization to your application • has a compiler start-up **profile** • supports direct access to MS-DOS; library supports DOS 3.X file-sharing • generates symbolic debugger information for use with all known MS-DOS debuggers • allows **exec-ing** subprocesses • was designed for professional software developers, not hobbyists • comes with great technical support by a company that specializes in compilers • comes with extensive **typeset documentation** • and *more*... call or write for your information packet today...

Not recommended for casual use, but for applications needing **industrial-strength tools**, contact

Meta



Ware™

INCORPORATED

903 Pacific Avenue, Suite 201, Santa Cruz, CA 95060-4429
(408) 429-6382 (429-META), TELEX: 4930879

Abroad:

ABC Software, The Netherlands
Microsoft, Tokyo
Grey Matter, United Kingdom
Buchdata, Frankfurt

Professional Tools Since 1979

High C V1.3: \$495 —on any MS/PC-DOS system—

OEMs: Contact us about porting our professional compilers to your systems.

TWS: Professional Compiler Developers and competitors, ask about our Translator Writing System compiler toolbox; see the review in *Computer Language*, December, 1985.

DOS Helper™: Need UNIX-like utilities to enhance MS-DOS? Ask about our powerful tools for \$49.95—including free with MetaWare compilers: FIND, TAIL, MV, LS, CAT, UNIQ, FGREP, and WC.

• MetaWare, High C, Professional Pascal, and DOS Helper are trademarks of MetaWare Incorporated • Other trademarks and their owners are: UNIX—AT&T, dBASE III—Ashton-Tate, Volkswriter Deluxe—Lifetree Software, GEM-DOS—Digital Research, Ada-DoD. © 1986 MetaWare.

MODULA-2

(continued from page 48)

modules to jump from 200 to 400. You generate map files by turning on the corresponding switch.

M2SDS

ITC's M2SDS uses pull-down menu windows and pop-up windows extensively. When users first enter M2SDS, it displays the Library-Tray and the Menu-Pick windows. The Library-Tray is used to recall packed libraries of modules. The Menu-Pick window offers five options: Desk, File, Tools, Edit, and Controls. The Desk contains a calculator, ASCII table, time display, and help. The File option is used to save an edited file, quit the editor without saving it, and exit back to DOS. The Tools option provides options to edit, compile, link, display a file's contents, read/write source code in ASCII form, and escape to DOS

(you return to M2SDS by typing *EXIT* in DOS). The Edit option allows you to mark, cut, copy, delete, and paste edited text. The Controls option enables you to resize and move a currently displayed window, as well as toggle the PC beeper.

The ITC compiler is integrated with the syntax-oriented editor, and you invoke it by pressing the Ctrl-\ key combination. A window appears displaying the compiler switches and their current status. The switches include those for use of 8087, stack, range, and arithmetic checking. If the compiler finds an error, it will display a diagnostic message and place the cursor at the first error location.

The compiler can remember 20 errors, which you can examine in sequence by pressing Ctrl-E.

The M2SDS linker can be invoked from the Edit/Compile/Link option box. The linker, which has no switches, produces .EXE files and can also link overlays.

PC Modula-2

The PC Modula-2 compiler is invoked from DOS by typing *modula* followed by an optional source file name. If the file name is omitted, the compiler will prompt for one. The compiler options are typed after the file name at either the DOS-level or the compiler prompt. At the end of program compilation, you are asked to type in another file name or exit. When the compiler detects errors, it displays diagnostic messages on the screen and in the listing file if generated.

The PC Modula-2 compiler has six switches. They control range, stack, and arithmetic overflow checking. In addition, they affect generating a listing file, query for symbol file names, and display information about the compiler version.

PC Modula-2 offers three formats for compiled programs: unlinked, linked, and executable. Unlinked object files contain code for the program itself and none for any module it uses. They are executed by invoking the

Vendors

Modula-2 PC

PCollier Systems
7925-A North Oracle Rd., Ste. 390
Tucson, AZ 85704
(800) 522-2060
\$59.95
Reader Service Number 37

M2SDS

Interface Technologies Inc.
3336 Richmond, Ste. 200
Houston, TX 77098
(713) 523-8422
\$80.88
Reader Service Number 38

Modula-2/86

Logitech Inc.
805 Veterans Blvd., Ste. 201
Redwood City, CA 94063
(415) 365-9852
\$89, base system
\$129, 8087 support system
\$189, 512K memory usage
Reader Service Number 39

PC Modula-2

Modula Corp.
950 North University Ave.
Provo, UT 84604
(801) 375-7400
\$150
Reader Service Number 40

HELP

NO
ROYALTIES

is at hand

HELP/Control™ - an online help subsystem for the IBM-PC.
Increase product marketability. Reduce product development time.

HELP/Control is fast. Screens - up to full display size - appear almost instantaneously because the runtime system is written in assembler and accesses the screen memory directly. Smaller screens appear as windows. Once they've helped, the original display is restored just as quickly.

HELP for the programmer. A few simple subroutine calls add a full-featured online help subsystem to your application because **HELP/Control** is a runtime system linked or loaded with that application. **HELP/Control** has been fully tested with Microsoft C, Lattice C, Turbo Pascal, IBM BASIC (Interpreter and Compiler), Microsoft FORTRAN, IBM COBOL and assembler. It is distributed with sample programs in each language.

HELP for the documentation writer. Build screens. Define captions. Associate each caption with a screen. **HELP/Control** includes a screen generation language, **HELPGEN**, that reads your sources, creates a runtime file, and lets you use your favorite editor. Our **HELP/Control** reads that file and displays help screens when the user activates the help function.

HELP for the end user. Use the cursor control keys to select a caption. Press return. That's it. Each screen has a number of highlighted captions which indicate other screens with information on that subject. Lotus 1-2-3 users will feel right at home. We've even made the entire **HELP/Control** manual available as a set of screens so you can browse without ever cracking a book.

HELP/Control comes complete with a detailed manual, both online and printed, with information for the programmer and documentation writer. It also includes instructions for the end user which may be incorporated into the application documentation.

PC-DOS 2.0 or greater required for developing **HELP/Control** applications. Applications using **HELP/Control** will run under PC-DOS 1.0 or greater. The runtime system requires approximately 9K for code and buffers for full size help screens.

The complete **HELP/Control** package (software, both manuals, demo programs) is \$125.00. A demonstration diskette, including the online manual, is available for \$15.00. To order or for more information, contact MDS, Inc. at (207) 772-5436. MasterCard and VISA accepted.

MDS, INC., P.O. BOX 1237, PORTLAND, MAINE 04104



Circle no. 285 on reader service card.

MODULA-2

(continued from page 53)

program loader, M2XA.EXE, which is responsible for an on-the-fly module linking and program execution. The second alternative is to use the linker LINKRLX for linking the object files of the compiled program and library modules employed. The program loader is still needed to run the linked file, but the loading operation is faster. The third alternative is to invoke the linker LINKEXE for linking the program object file with the library modules and run-time support, producing a stand-alone .EXE file. PC Modula-2's two linkers operate similarly to the way in which the compiler does. LINKRLX has one directive—namely, the query option—whereas LINKEXE has no directives.

Modula-2PC

The Modula-2PC compiler is invoked from DOS by typing *m2pc* followed by an optional program source file name and compiler switches. If the file name is omitted at the DOS level, the compiler prompts you for one. There are four compiler switches. They control range checking (for array indices and verification of valid integer and long integer values), nil pointer check, and conversion of source file text into uppercase. The fourth switch is used to compile a single file and exit to DOS. Normally, the compiler prompts for another file once the current one is processed. Error messages are sent to a .LST file and are not displayed on the screen.

Modula-2PC provides M2X.EXE as a run-time environment. You can invoke its sole directive to create a stand-alone .EXE file. The manual states that there is no link step. Thus M2X.EXE has a dual role—it's used either to run your compiled program or to produce an .EXE file and stop.

Editors

Modula-2/86

The MOD editor that Logitech offers to go along with its Modula-2/86 compiler and linker has several interesting features. It operates as a free-form editor with optional macros for program constructs. Moreover, it supports windows and a mouse and becomes the core of an integrated system. You can invoke the compiler

	Modula-2/86	M2SDS	PC Modula-2	Modula-2PC
Version	2.0	2.00	1.0	1.0
Compiler	yes	yes	yes	yes
Number of passes	4	incremental	1	1
Linker	yes	yes	yes	no
Editor	yes(opt)	yes	no	no
Syntax-oriented	no	yes	n/a	n/a
Produce M-code	no	no	no	no
Produce native code	yes	yes	yes	yes
Optional post-mortem debugger	yes	no	no	no
Optional run-time debugger	yes	yes	yes	no

Table 1: General comparisons

	Modula-2/86	M2SDS	PC Modula-2	Modula-2PC
87 support	yes	yes ¹	yes	no
Absolute variables	yes	yes	yes	yes
Address	yes	yes	yes	yes
Array	yes	yes	yes	yes
Bit set	yes	yes	yes	yes
Boolean	yes	yes	yes	yes
Byte	yes	yes	no	no
Cardinal	yes	yes	yes	yes
Character	yes	yes	yes	yes
Enumerated	yes	yes	yes	yes
Integer	yes	yes	yes	yes
Long integer	no	yes	yes	yes
LongSet library support	no ²	yes	yes	no
Bytes in REAL	8	8	4	8
Export opaque data	yes	yes	yes	yes
Pointer	yes	yes	yes	yes
Proc. type	yes	yes	yes	yes
Procedural parameter	yes	yes	yes	yes
Process	yes	yes	yes	yes
Real	yes	yes	yes	yes
Record	yes	yes	yes	yes
Set	yes	yes	yes	yes
Subrange	yes	yes	yes	yes
WORD	yes	yes	yes	yes

¹ Tests showed support severely malfunctioning.

² Available with the Logitech Translator for Turbo Pascal programs

Table 2: Data types

	Modula-2/86	M2SDS	PC Modula-2	Modula-2PC
Assembly-language interface	yes	yes	yes	yes
Chaining	yes	yes	yes	yes
Concurrent processes	yes	yes	yes	yes
Cursor/screen control	no	yes	no	yes
DOS calls	yes	yes	yes	yes
High-res graphics	no	yes	yes	no
In-line code	yes	yes	no	yes
Interrupts	yes	yes	no	no
Mouse interface	yes	yes	yes	no
Overlays	yes	yes	yes	yes
Time/date	yes	yes	yes	yes

Table 3: Programming features

The Peak of Performance

SCALE THE HEIGHTS OF PRODUCTIVITY

Sure, you've proven that in your hands a computer is a productive tool. But if you haven't teamed up with a SemiDisk you have heights yet to climb!

IT'S NO MERE RAMDISK

SemiDisk has been leading the way for Disk Emulators since their inception. If you've seen RAMdisks you know what it's like to load programs in an

instant, and read or write files without delay. Unlike alternatives, the SemiDisk offers up to 8 megabytes of instant-access storage while leaving your computer's main memory free for what it does best - computing!

KEEP A GRIP ON DATA

Go ahead, turn off your computer. Take a vacation. With the battery backup option, your valuable data will be there in the morning even if you aren't. You'll sleep better knowing not even a 5 hour blackout will sabotage your files.

SEMI

SemiDisk Systems, Inc.

P.O. Box GG, Beaverton, Oregon 97075

503-626-3104

NEW LOWER SEMIDISK PRICES THAT WON'T SNOW YOU UNDER

	512K	2Mbyte
IBM PC, XT, AT	\$495	\$995
Epson QX-10	\$595	\$995
S-100, SemiDisk II	\$799	\$1295
S-100, SemiDisk I	\$595	—
TRS-80 II, 12, 16	\$695	\$1295
Battery Backup Unit	\$130	\$130

Software drivers available for CP/M 80, MS-DOS, ZDOS, TurboDOS, and VALDOCS 2

MODULA-2

(continued from page 54)

or linker or run any program from within the editor and return to it upon either completion or interruption. When compilation errors occur, the MOD editor remembers them and allows you to view the compiler error messages as a circular list of messages appearing inside a window.

The Emacs-like editor is very easy to use. You use function keys to display the main menu (which appears as a pop-up window) or to carry out a variety of tasks, such as calling an on-

line help screen for the MOD commands, loading and saving files, calling the compiler or linker, and invoking a very fast syntax checker. The latter verifies the basic Modula-2 syntax; it does not detect misspelled variables, undeclared identifiers, and so on. In other words, the MOD syntax checker does not duplicate the compiler's task of verifying the correctness of the source code, but it is useful for detecting missing colons, semicolons, and other Modula-2 keywords.

MOD supports multiple windows that can be opened vertically or horizontally. Adjusting window shapes is

also supported. The F7 function key lets you navigate from one window to another. The editor also has search and find/replace features. You can search downward or upward starting at the current cursor position. When finding/replacing text, prompts ask you if this should be done in automatic or query modes. Although the line and column positions of the cursor are not displayed constantly, pressing Alt-F3 reveals that information. The Alt-F4 key combination allows you to reposition the cursor at the beginning of a specified line.

You move a block of text by marking the block, deleting it into a buffer, and then inserting it at the new location. For WordStar users this seems a bit different and somewhat dangerous because deleting more text clears the buffers of any previous contents. The editor would be improved if markers could be used directly for moving and copying text. Text can also be moved between windows.

I mentioned earlier that the MOD editor has special macros for program constructs. Pressing Alt-F2 causes a special configuration file to be read assigning macros to alternate keys. If you use keyboard macro programs such as SuperKey, you should clear the keyboard definition to make way for the MOD assignments. Pressing Alt-letter will result, in most cases, in the appearance of some construct related to the letter—for example, Alt-W causes a *while-do* loop to appear and the cursor is placed between the *while* and *do* keywords. The beauty of MOD is that you are still free to move around.

M2SDS

The M2SDS syntax-oriented editor is among the most flexible syntax-oriented editors I have tested. You have to develop a reflex to use this type of editor. It provides you with syntactically arranged placeholders that are based on Modula-2 syntax. The placeholders are filled by assigning declarations to them using Alt key combinations—for example, to declare constants you press Alt-C, to declare variables you press Alt-V, and so on. These declarations insert Modula-2 keywords and other placeholders. For example, to create an import list, you type Alt-I, and the editor inserts

	Modula-2/86	M2SDS	PC Modula-2	Modula-2PC
ASCII files	yes	yes	yes	yes
Binary files	yes	yes	yes	yes
Untyped files	no	no	no	no
Sequential	yes	yes	yes	yes
Manipulate file position	yes	yes	yes	yes
Access file directory	yes	yes	yes	yes
Manipulate file directory	yes	yes	yes	yes

Table 4: File I/O library support

	Modula-2/86	M2SDS	PC Modula-2	Modula-2PC
SYSTEM Module				
PROCESS type	yes	no	yes	yes
NEWPROCESS	yes	yes	yes	yes
TRANSFER	yes	yes	yes	yes
IOTRANSFER	yes	yes	no	yes
Process Module				
SIGNAL data type	yes	no	yes	no
StartProcess	yes	no	yes	no
SEND	yes	no	yes	no
WAIT	yes	no	yes	no
Awaited	yes	no	yes	no
Init	yes	no	yes	no

Table 5: Concurrency features

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)
Modula-2/86	768	27728	4 pass 1:23 M2C.EXE 0:54	00:18
M2SDS	768	21370	0:31	00:17
PC Modula-2	768	42064	0:24	00:23
Modula-2PC	768	41472	0:14	00:35
Turbo Pascal	768	11495	0:02	00:13

Table 6: Results of the sieve benchmark test

10 MHz 286 TURBO™



The Code of the West

The Good

Western 286 Turbo™

Western Computer Corp. has built a strong reputation in the Micro-computer Industry by sticking to a few basic principles. We call it the "Code of the West" and it goes something like this:

MAKE IT BETTER.

with added features like 8-10 MHz switchable, 512K of RAM expandable to 1MB, Standard RS-232, Parallel Ports and 1 year Warranty on parts and labor.

MAKE IT NOW.

it doesn't matter how good a deal you make if you have to wait forever to get it!

MAKE IT CHEAPER.

and pass the savings on to the Buyer.

MAKE IT COMPATIBLE.

we can't think of anything worse than getting a good deal on a new system, but not being able to run your favorite Software! By the way, we use Phoenix Bios for total compatibility.

MAKE IT IN AMERICA.

Please!

The Bad

\$5295⁰⁰

What could a Personal Computer with fewer standard features, actually do for you that would compel you to part with an additional \$3,300.00. Will it pick-up the kids from school, wash the car or walk the dog? Think about it!

& The Ugly

Regardless of what you pay for it, a Foreign Import can be one big gamble, with no guarantees of compatibility, service, or support.

Western Computer Corp. has a full line of personal computers and peripherals designed around your needs and your budget including: **WESTERN COMPUTER XT TURBO™**, **WESTERN CARRY-ON™** and...

...Coming soon—**WESTERN 80386™**

Call one of our salespersons at **(714) 553-1611** to find out how to hang-on to more of your money!

Western Computer

17781 Mitchell St., Irvine, CA 92714 USA
Phone (714) 553-1611
Customer Service Only (714) 533-1705
Telex: 7566731
Answer Back WESTERN COMP
FAX (714) 553-0236

Western Computer Australasia Limited
4-82 Abbot Street, Ascot Brisba
Queensland, Australia 4007
*(07) 268-6589 - Telex: AA144746
FAX: *(07) 2685256 - Answer Back MCGUIR

Western Computer Spain
Diputacion, 238-6° Despacho 8
08007 Barcelone, Spain
Tele: 317 7128

Western Computer Greece - Busisoft Ltd.
102 Syngrou Ave., Athens 11741, Greece
Tele: 902 4248 - TLX: 210364

FROM <id> IMPORT <id-list>. You then fill in the <id> and <id-list> placeholders. The editor provides placeholders for declarations, procedural parameters, data types, and statements.

Cursor movement is achieved by using the four arrow keys, Home, End, and PgUp. The latter is used to move up one program level. The editor also provides the ability to search and replace program text, scanning forward or backward. You can scan tokens (that is, complete identifiers), strings, and declarations. You can copy and move single or multiple statements.

The editor has an interesting feature that allows you to transform a construct. This is very useful in correcting or editing a program—for example, you can transform a *for* loop with no *step* clause into one that has it. A second example is converting a procedure into a function by append-

ing a returned value declaration.

Language Features

All the Modula-2 implementations support the basic data types and the imported types *WORD* and *ADDRESS*, defined by Wirth in the second edition of *Programming in Modula-2* (Berlin: Springer-Verlag, 1983). The long integer *LONGINT*, introduced in Wirth's third edition of the language reference book, is supported by the M2SDS, PC Modula-2, and Modula-2PC compilers. The *BYTE* type is supported by Modula-2/86 and M2SDS. All Modula-2 implementations except Modula-2PC support the 8087 chip. As mentioned earlier, PCColliers Systems intends to offer 8087 support later this year.

M2SDS defines the type *STRING* in its *SYSTEM* module. Variables that are defined as strings use *STRING(n)*, where *n* is the string length varying from 1 to 255. The string length is stored at index 0. *STRING*-typed variables are compatible with *ARRAY OF CHAR* (open arrays of characters) used

as parameters in procedures. All the other Modula-2 implementations follow the standard language and treat strings as arrays of characters. All implementations support procedural and functional types.

All the Modula-2 packages implement the standard Modula-2 loops and decision-making constructs. Logitech's Modula-2/86 follows Wirth's second edition of the language reference book.

User-defined procedures and functions also follow the standard language definition. Modula-2PC implements the Pascal *FORWARD* statement.

Libraries

Modula-2 is a small core language that relies heavily on library modules for extensions. This includes modules that export additional data types, file I/O routines, screen handling, low-level access, and so on.

All the Modula-2 implementations export the *WORD* and *ADDRESS* data types from the pseudomodule *SYS-TEM*. Modula-2/86 and M2SDS also export the *BYTE* type from *SYSTEM*. Modula-2/86 supports 18-digit decimals through the *Decimals* module, which provides arithmetic operations, string-to-decimal conversion, and error checking. In addition, M2SDS exports the *STRING* type discussed earlier and provides a *String* module that manipulates strings and arrays of characters. The other Modula-2 implementations offer string-handling modules that tackle arrays of characters. PC Modula-2 and the extended library of M2SDS (that is, that of SDS-XP) provide a module for *LongSets* with set membership size exceeding the dismal standard of 16 members. Logitech is providing a similar module with its Translator that converts Turbo Pascal programs to Modula-2/86 programs. All the Modula-2 vendors include data-conversion modules to convert numeric data types (integers, long integers, cardinals, and reals) into strings and vice versa.

Modula-2/86 and M2SDS include the standard Wirth version of the math function library *MathLib0*. PC Modula-2 offers its *MathLib1* library that includes a large set of functions, such as a complete set of trigonometric functions and their inverses, base-10 log, hyperbolic, power, and magnitude functions. PC Modula-2 includes

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)	Comments
Modula-2/86	1536	35696	4 pass 1:50		no 8087
			M2C.EXE 1:21	00:21	no 8087
		33040	4 pass 1:39		with 8087
			M2C.EXE 1:13	00:07	with 8087
M2SDS	1536	30884	0:37	00:22	no 8087
		30852	0:39	error	with 8087
PC Modula-2	1536	63424	0:29	error	no 8087
		62896	0:29	error	with 8087
Modula-2PC	1536	39936	0:15	00:17	no 8087
Turbo Pascal	1792	12245	0:02.5	00:15	no 8087
		11072	0:02.4	00:06	with 8087

Table 7: Results of the sort benchmark test

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)
Modula-2/86	2304	28224	4 pass 1:26	
			M2C.EXE 1:01	00:37
M2SDS	2304	21834	0:33	00:34
PC Modula-2	2304	42781	0:26	00:40
Modula-2PC	2304	30720	0:13	01:24
Turbo Pascal	2045	11886	0:02.4	00:25

Table 8: Results of the matrix-inversion, floating-point test

dBASE and Pascal. Together at last.

It had to happen.

They've been going together for years, but always with a lot of "glue, tape and hope" procedures that kept it from being a perfect union.

Announcing a new dBASE[®] enhancement tool that includes an interface to Turbo Pascal[™] and a Pascal library. So you can customize the dBASE standard to your special business and scientific/engineering needs.

Array handling.

Random Number
Generation function.

Variance and coefficient
of variance.

Standard deviation
function.

Skewness and sample
kurtosis.

And business functions
like present and future
value, net present value, loan

amortization, and internal rate
of return.

Plus you'll be able to write
and store custom functions
in the library.

To order, or for more
information or the name of
your nearest dealer, call the
Ashton-Tate[®] Publishing Group
at (800) 437-4329, Ext. 227.

And live happily ever
after.



Trademarks/owner: dBASE, Ashton-Tate/Ashton-Tate, Turbo Pascal/Borland International. © 1986 Ashton-Tate. All rights reserved.

Circle no. 242 on reader service card.

MODULA-2

(continued from page 58)

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)	Comments
Modula-2/86	2432	36448	4 pass 1:54 M2C.EXE 1:29	00:54 02:11 01:55 02:10 01:25	no 8087 SQRT() LN() EXP() ARCTAN() SIN()
	2432	32240	4 pass 1:48 M2C.EXE 1:24	00:08 00:09 00:28 00:11 00:12	with 8087 SQRT() LN() EXP() ARCTAN() SIN()
M2SDS	2432	36870	0:39	03:28 05:36 05:25 05:44 05:13	no 8087 SQRT() LN() EXP() ARCTAN() SIN()
	2432	24952	0:35	error error error error error	with 8087 SQRT() LN() EXP() ARCTAN() SIN()
PC Modula-2	2432	72864	0:32	01:52 01:51 02:03 01:56 01:47	no 8087 SQRT() LN() EXP() ARCTAN() SIN()
	2432	61520	0:31	00:12 00:15 error 00:11 00:12	with 8087 SQRT() LN() EXP() ARCTAN() SIN()
Modula-2PC	2432	39424	0:15	36:46 27:43 08:01 28:39 06:49	no 8087 SQRT() LN() EXP() ARCTAN() SIN()
Turbo Pascal	1947	12607	0:02.6	01:41 02:39 02:17 02:23 01:56	no 8087 SQRT() LN() EXP() ARCTAN() SIN()
		11482	0:02.6	00:08 00:09 00:12 00:10 00:11	with 8087 SQRT() LN() EXP() ARCTAN() SIN()

Table 9: Results of mathematical-functions benchmark test

a second module to provide mathematical constants and the range of reals. Modula-2PC includes similar extensions.

Console I/O is generally supported by the *InOut* and *RealInOut* modules in standard Modula-2. Each implementation offers a superset of this standard—Modula-2/86, PC Modula-2, and Modula-2PC include *WORD* I/O routines; M2SDS adds *STRING* and *Record* I/O procedures. PC Modula-2 offers routines for formatted output of *REAL* numbers and for displaying them as octal numbers.

M2SDS and Modula-2PC provide screen/cursor-control modules. M2SDS and PC Modula-2 supply high-resolution color-graphics modules. PC Modula-2 includes support for using a mouse in graphics mode. Modula-2/86 has no modules for any of these operations.

Each Modula-2 package provides a collection of routines for file I/O. The Modula-2 vendors have used the standard Wirth modules as a starting point and added more procedures and new modules to tap into the routines of PC-DOS/MS-DOS that perform file, directory, and drive manipulation. Modula-2/86, M2SDS, and PC Modula-2 provide many additional and practical I/O routines, including filename query, path manipulation, and byte-block I/O.

Accessing PC-DOS/MS-DOS routines and the 8088 CPU registers opens the door for you to use more hardware and operating-system capabilities. The four Modula-2 packages permit such access. With the exception of PC Modula-2, the Modula-2 implementations allow in-line code to be inserted in the source program.

All the implementations support concurrency. Only Modula-2/86 and PC Modula-2, however, provide the *Process* module, which is used in concurrent-process synchronization. M2SDS implements concurrent processes slightly differently from standard Modula-2.

Benchmarks

I ran the following benchmarks using an IBM PC/XT with 512K RAM, an 8087 chip, 20-megabyte hard disk, and PC-DOS/MS-DOS (3.1). The benchmark pro-

grams were taken from popular tests or common operations, such as sorting and matrix inversion. I used two types of timing schemes, depending on the test program. In the first method, a timer was turned on after the program name was typed at the DOS level and turned off once the program ended and the DOS prompt reappeared. The second method used internal prompts to start and stop timing. The tests I used were:

- Sieve test—the popular sieve of Eratosthenes, which finds prime numbers. The upper limit of the prime numbers is 7,000.
- Integer sort test—measures the speed of manipulating integer arrays. A 1,000-member array is created in order and reverse-sorted ten times using the Shell-Metzner sort method.
- Matrix-inversion test—measures the speed of basic floating-point operations. A square matrix with 20 rows and columns is created by assigning 1s to all nondiagonal elements and 2s to all the diagonal elements. The test was carried out once without 8087 support and once with it.
- Mathematical-functions test—measures the speed of the square root, natural logarithm, exponential, arctangent, and sine functions. The arguments of each function are varied. The tests were carried out once without 8087 support and once with 8087 support.
- Disk-write test—measures the speed of writing 512 blocks of 128 characters to a text file stored on a floppy disk. Initially, the disk is empty.
- Disk-read test—measures the speed of reading 512 blocks of 128 characters from a text file stored on a floppy disk. The disk contains no other files.
- Recursive quicksort test—measures the speed of recursion. A 1,000-integer array is created in order and reverse-sorted using a recursive quicksort. The above process is repeated ten times.
- Dynamic-allocation and pointer test—measures the speed of dynamically allocating a binary tree and the speed of accessing the tree elements using pointers. The test program creates an array of 1,000 integers using truncated sine-function values.

All the Modula-2 compilers were made to produce stand-alone .EXE

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)	Comments
Modula-2/86	768	27568	4 pass 1:31 M2C.EXE 1:06	01:25	
M2SDS	768	21258		0:31	09:01
PC Modula-2	768	41888		0:25	01:49
Modula-2PC	768	27648		0:13	06:28
Turbo Pascal	512	11478		0:02	00:57 Buffered Pascal output

Table 10: Results for disk-write benchmark test

PolyMake The Leading Make Utility

Are you still using a prehistoric Make? Now, step up to PolyMake, the most powerful and flexible Make utility available for programmers using MS-DOS. PolyMake is like an intelligent assistant that remembers how to rebuild a program when you change one part of the program. PolyMake will automatically invoke your compiler, assembler, linker, librarian or other tools to update a single program or entire software systems when you type — MAKE. PolyMake comes with built-in rules for rebuilding programs but you can also teach it new rules so you don't have to remember the file dependencies in your program. Advanced programmers prefer capabilities like fully recursive makefile processing and the ability to invoke PolyMake with special "flags" and macros that automate a whole range of tasks. New Make users appreciate the Step-By-Step tutorial and intuitive commands. Handles source files written in any language. Requires DOS 2.0 & higher. Compatible with LANs, the IBM PC, XT, AT and other MS-DOS PCs. For complete details write for the POLYTRON Programmer's Catalog. **\$99**

PVCS The Most Powerful & Flexible Source Code Revision & Version Control System.

The POLYTRON Version Control System (PVCS) allows programmers, project managers, librarians and system administrators to effectively control the proliferation of revisions and versions of source code in software systems and products. PVCS is a superb tool for programmers and programming teams. (A special LAN version is also available.) If you allow simultaneous changes to a module PVCS can merge the changes into a single new revision. If changes conflict, the user is notified. Powerful capabilities include: Stores and retrieves multiple revisions of text; Maintains a complete history of revisions to act as an "audit trail" to monitor the evolution of a software system; Maintains separate lines of development or "branching"; Provides for levels of security to assure system integrity; Uses an intelligent "difference detection" to minimize the amount of disk space required to store a new version. Requires DOS 2.0 or higher. Compatible with the IBM PC, XT, AT and other MS-DOS PCs. Single User version \$395. 5-station LAN version \$1,000, add \$500 for each additional 5 stations. **\$395**

TO ORDER: VISA/MC 1-800-547-4000, Dept. No. 355; Oregon/Outside US, 503-684-3000
Send Checks/POs To: POLYTRON Corp. 1815 NW 169th Pl., #2110, Dept. No. 355, Beaverton, OR 97006

POLYTRON
High Quality Software Since 1982®

Circle no. 283 on reader service card.

MODULA-2

(continued from page 61)

files, as opposed to intermediate compiled programs that run within Modula shells.

The Modula-2PC index and pointer-

checking directives were turned on during program compilation because all the other compilers that were tested had their checking switches on by default. Batch files were used in compiling and linking the test programs that I have listed above.

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)	Comments
Modula-2/86	693	27552	4 pass 1:31 M2C.EXE 1:06	00:57	
M2SDS	768	21242	0:32	07:36	
PC Modula-2	693	41840	0:24	01:22	
Modula-2PC	768	27648	0:13	04:54	
Turbo Pascal	384	11355	0:02	00:30	Buffered Pascal input

Table 11: Results for disk-read benchmark test

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)	
Modula-2/86	1920	28080	4 pass 1:29 M2C.EXE 1:04	00:11	
M2SDS	1920	21690	0:27	00:09	
PC Modula-2	1920	42672	0:26	00:12	
Modula-2PC	1920	30720	0:14	00:19	
Turbo Pascal	1685	11822	0:03	00:07	

Table 12: Results for recursion benchmark test

Compiler	Source (bytes)	EXE	Compile + Link (mm:ss)	Run Time (mm:ss)	Comments
Modula-2/86	2816	34720	4 pass 1:46 M2C.EXE 1:24	00:24 00:08	allocate search
M2SDS	2816	36086	0:39	01:55 00:05(err)	allocate search
PC Modula-2	2816	71984	0:32	01:05 00:20	allocate search
Modula-2PC	2816	43520	0:15	01:29 00:19	allocate search
Turbo Pascal	2521	12216	0:03	00:22 00:06	allocate search

Table 13: Results for dynamic-allocation and pointer test

Compiler	Average Index	Standard Deviation	Sample Size
Modula-2/86	1.047	0.082	19
M2SDS	3.317	2.225	12
PC Modula-2	1.462	0.531	16
Modula-2PC	7.930	10.872	13

Table 14: Statistics for relative run-time speed indices

Bugs

I encountered several bugs while carrying out the benchmark testing. They were:

- The Modula-2PC compiler did not accept array indices that were *INTegers*. The loop control variables were changed into *CARDINALs* for the sieve and matrix-inversion tests.
- The matrix-inversion test program compiled by PC Modula-2 exhibited a run-time error with and without the 8087 support. In the first case, a "CARDINAL OVERFLOW" error message appeared; in the second, a "REAL OVERFLOW" error message was displayed. I informed Modula Corp. about this bug.
- The 8087 support of M2SDS did not function properly. I experienced run-time errors with all the benchmark programs compiled with the 8087 support turned on. I wrote additional programs to further test and inspect the proper function of the basic four floating-point operations and the math functions used in the benchmarks. The results showed a severe malfunction of the 8087 support. I informed ITC of this malfunction.
- PC Modula-2 had a bug with the exponential function when used with 8087 support. All valid arguments supplied to this function returned an overflow message.
- The disk-write program compiled by Modula-2PC hung the system. I contacted PColliers Systems and was asked to remove the *Reset()* statement in the program. The recommendation worked, and I was able to run the benchmark.
- The square root function of Modula-2PC hung the system when given the argument zero (0.0). I had to modify the benchmark so that the square root argument varied from 1 to 1,001. I contacted PColliers Systems and reported the bug and was later informed that it was fixed.
- Running the dynamic-allocation and pointer test with M2SDS, I noticed that the pointer-access test was unusually fast. Inserting some *WriteString('')* statements, I was able to detect some shortcircuiting in the program flow.

Test Results

The results of the benchmark tests are shown in Tables 6–13, pages 56–62. I have included results from running

Turbo Pascal with Pascal versions of the benchmarks. The main purpose is to compare run-time speed and not compilation speed or code size. Notice that the Turbo Pascal programs run faster than any compiled Modula-2 program except for the mathematical functions when no 8087 support is used. This should answer a lot of questions about comparing the speed of Turbo Pascal programs with those produced by various Modula-2 compilers.

M2SDS produces the smallest files, closely followed by Modula-2/86, then Modula-2PC and PC Modula-2. Looking at the data for the speed of producing .EXE files, you can see that Modula-2PC is the clear winner. It is followed by PC Modula-2. M2SDS is the third in rank, and Modula-2/86 lags on the average by 1 minute and 10 seconds behind Modula-2PC.

Looking at run-time speed data, you can see that different implementations may shine when running different tests. M2SDS comes first in the sieve and integer-sort tests. Modula-2PC does well with the matrix-inversion test. Logitech's Modula-2/86 is the only one that runs the same test with 8087 support.

Looking at the math-functions test with no 8087 support, you can see that Modula-2/86 and PC Modula-2 compete for first place, M2SDS comes third, and Modula-2PC a distant fourth. The data for the same test running with 8087 support shows a close timing between Modula-2/86 and Modula-2PC.

The disk-write and -read tests reflect the use of faster buffered I/O by Modula-2/86 and PC Modula-2. In writing to the disk, Modula-2PC came third and M2SDS fourth. The last two compilers reverse their ranks in the disk-read test.

The recursive quicksort test places M2SDS in the lead, followed by Modula-2/86 and with PC Modula-2 a close third. Modula-2PC took twice as long to finish the test as did M2SDS.

In the dynamic-allocation test, Modula-2/86 performed very well and was followed by PC Modula-2 and Modula-2PC. M2SDS took the longest time.

To give you a general idea of overall run-time speed, I present some basic statistics in Table 14, page 62. The calculations are based on the following conditions:

1. For each test, divide the run-time values by the least observed value. This gives a relative run-time speed index.
2. The use of 8087 support is regarded as a different set.
3. Run-time errors creating missing data are ignored.
4. For each compiler, I calculated the average and standard deviations of the relative run-time speed values.

Table 14 shows that Modula-2/86

benchmark programs have the fastest run times overall, followed closely by PC Modula-2. The standard deviation gives you an idea of the spread in speed. Notice that Modula-2PC has a standard deviation greater than its average value, indicating that it did very well on some tests and lagged behind on others.

DDJ

Vote for your favorite feature/article.
Circle Reader Service No. 5.

C-terp

The C
Interpreter
You Won't
Outgrow



C-terp will grow with you as you progress from novice through professional to guru. Unbelievable, but true, the easiest-to-use C interpreter will provide you with the most advanced programming features for upward growth. Our exclusive **object module support** enables you to add libraries (like HALO, PANEL, Windows for C, etc., or your own homebrew libraries) to C-terp as you add them to your computing repertoire. Use C-terp as a microscope on your libraries! Flip a bit and allow our **software paging** (NEW) to handle those big jobs! There are no fixed-size tables to overflow, and C-terp can be configured for different screens and screen adapters (NEW). With multiple modules and **full K&R support**, we offer a dream C environment.

- Our new improved **configurable editor** competes with anything going.
- Speed -- Linking and semi-compilation are breathtakingly fast.
- Convenience -- Errors direct you back to the editor with the cursor set to the trouble spot.
- Symbolic Debugging -- Set breakpoints, single-step, and directly execute C expressions.
- Compatibility guaranteed -- batch file to link in your compiler's entire library. Supported compilers include: **Computer Innovations C86, Lattice C, Microsoft C 3.0, Mark Williams C86, and Aztec C.**
- Many more features including batch mode and 8087 support.

What Our Users/ Reviewers Are Saying

- "... easy to use, powerful, and a timesaver."
- "... we absolutely LOVE C-terp."
- "... has restored my faith in interpreters."
- "... a programmer's dream."
- "... wonderful technical assistance."
- "... increased our productivity by a factor of 40."
- "... the best C product ever, in any category."

- Price: \$300.00 (Demo \$45.00)
MC, VISA

Prices include documentation and shipping within U.S. PA residents add 6% sales tax. Specify compiler.

- C-terp runs on the IBM PC (or any BIOS compatible machine) under DOS 2.x and up with a suggested minimum of 256 Kb of memory. It can use all the memory available.

* C-terp is a trademark of Gimpe! Software.

GIMPE! SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261

Listing One (Text begins on page 22.)

Listing 1 -- more.c

```

1 #include <stdio.h>
2 #include <ctype.h>
3 #include <fcntl.h>
4 #include <process.h>
5
6 /*      MORE.C      Page input to stdout.
7 *
8 *      (C) 1986, Allen I. Holub. All rights reserved.
9 *
10 *      Usage: more [-<offset>] file...
11 *
12 *      Exit status: 0 always
13 */
14
15 /*-----*/
16
17 extern int  b_getc ();          /* Direct console input function */
18                                     /* source in /src/tools/b_getc.c */
19 extern int  look ();           /* Char. lookahead function. */
20                                     /* source in /src/tools/look.asm */
21 extern long  ftell ( FILE * ); /* Standard library functions: */
22 extern long  atol ( char* );
23 extern FILE  *fopen ( char*, char* );
24 extern int   fclose ( FILE* );
25 extern int   spawnl ( int, char*, char*, );
26 extern char  *getenv ( char* );
27 extern char  *fgets ( char*, int, FILE* );
28 extern long  filelength ( int );
29
30 /*-----*/
31
32 #define CAN      0x18 /* ^X */
33 #define ESC      0x1b /* ^[ */
34
35 #define max(a,b) ((a) > (b) ? (a) : (b))
36 #define min(a,b) ((a) < (b) ? (a) : (b))
37
38 #define BSIZE     256 /* Maximum length of a line in the file */
39 #define PAGESIZE  23  /* # of lines to output before stopping */
40
41 #define E(x)      fprintf(stderr, "%s\n", x)
42 #define HAS_DOT(p) strchr(p, '.')
43
44 FILE  *Ifile      = stdin; /* Current input file */
45 char  *Ifile_name = "/dev/con"; /* Name of file associated w/ Ifile */
46 int   Repeat_count = 1; /* Repeat count for most recent cmd */
47 long  Line         = 0; /* # of output lines printed so far */
48 long  Flen         = 0; /* Length of input file in chars */
49 long  Start_here   = 0; /* Seek to here when prog starts */
50
51 /*-----*/
52 * Stack used to keep track of start of lines. Maximum number
53 * of lines is determined by STACKSIZE.
54 */
55
56 typedef long  STACKTYPE;
57 #define STACKSIZE (1024*6) /* Must be divisible by 2 */
58
59 STACKTYPE Stack[ STACKSIZE ];
60 STACKTYPE *Sp = Stack + STACKSIZE;
61
62 #define STACKFULL (Sp <= Stack)
63 #define STACKEMPTY (Sp >= Stack + STACKSIZE)
64 #define CLEAR_STACK() Sp = Stack + STACKSIZE;
65 #define TOS (STACKEMPTY ? 0 : *Sp)
66 #define BACK_SCRN *( min( Sp+(PAGESIZE-1), Stack+(STACKSIZE-1) ) )
67
68 #define erase_line() line( ' ', 0 ) /* Draw a line of spaces */
69
70 /*-----*/
71
72 help()
73 {
74     register int  i;
75
76     /* Print a help message with a box around it, special IBM graphics
77      * characters are used for the box
78      */
79
80     putc( 0xd6, stderr );
81     for( i = 56; --i >= 0; putc( 0xc4, stderr ) )
82     {
83         E("\267");
84         E("\272 b ..... go (B)ack a page \272");
85         E("\272 e ..... go to end of file \272");
86         E("\272 n ..... go to (N)ext file \272");
87         E("\272 o ..... print (O)ffset from start of file \272");
88         E("\272 q ..... (Q)uit (return to DOS) \272");
89         E("\272 s ..... (S)kip one line (w/o printing) \272");
90         E("\272 r ..... (R)ewind file (go back to beginning) \272");
91         E("\272 l ..... execute a program (type blank line \272");
92         E("\272 ..... at prompt to execute last) \272");
93         E("\272 / ..... search for regular expression \272");
94         E("\272 ..... (type blank line at prompt for last) \272");
95         E("\272 ESC ..... Scroll until any key is hit \272");
96         E("\272 CR ..... next line \272");
97         E("\272 SP ..... next screen \272");
98         E("\272 anything else ... print this list \272");
99         E("\272 \272");
100        E("\272 All commands may be preceded by a count. \272");
101    }
102    putc( 0xd3, stderr );

```

(continued on page 66)

Clipper is the fastest dBase III and dBase III Plus™ compiler available. Nothing else comes close. When performance counts, experts rely on Clipper for more speed, more power, and more creative freedom. You can, too. Call for details.

- Clipper compiled programs run 2 to 20 times faster.
- No royalties...no runtime fees.
- Source code security.
- User defined functions.
- Arrays.
- Simple menu commands.
- Context sensitive help can be included with programs.
- More fields; more memory variables.
- Call C and Assembly programs.
- Complete debugging facilities.
- Multiple file relationships.
- IBM PC, XT, AT, 3270 compatible™.
- Multi-user capability.

Clipper™

CLIPPER. THE dBASE COMPILER.
A WINNING PERFORMANCE EVERY TIME.



Nantucket™

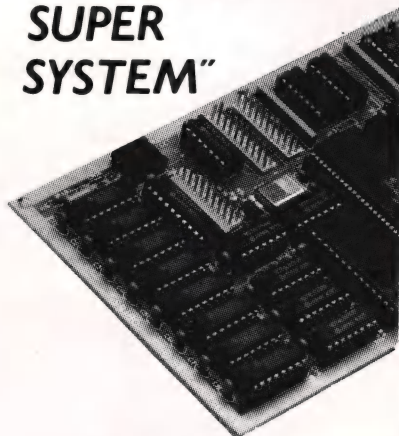
Nantucket Corporation
5995 South Sepulveda Boulevard
Culver City, California 90230
(213) 390-7923
Outside California call toll-free:
1-800-251-8438

dBase, dBase III, and dBase III Plus are trademarks of Ashton-Tate, Inc.
IBM PC, XT, AT, and 3270 are trademarks of International Business Machines Corporation.
Clipper and Nantucket are trademarks of Nantucket Corporation.

Circle no. 220 on reader service card.

Byte Magazine called it.

"CIARCIA'S SUPER SYSTEM"



The SB180 Computer/Controller

Featured on the cover of Byte, Sept. 1985,
the SB180 lets CP/M users upgrade to a
fast, 4" x 7 1/2" single board system.

- **6MHz 64180 CPU**
(Z80 instruction superset), 256K RAM,
8K Monitor ROM with device test, disk
format, read/write.
- **Mini/Micro Floppy Controller**
(1-4 drives, Single/Double Density,
1-2 sided, 40/77/80 track 3 1/2", 5 1/4"
and 8" drives).
- **Measures 4" x 7 1/2"**, with mounting holes
- **One Centronics Printer Port**
- **Two RS232C Serial Ports**
(75-19,200 baud with console port
auto-baud rate select).
- **Power Supply Requirements**
+5V +/-5% @500 mA
+12V +/- 20% @40mA
- **ZCPR3 (CP/M 2.2/3 compatible)**
- **Multiple disk formats supported**
- **Menu-based system customization**

SB180-1
SB180 computer board w/256K
bytes RAM and ROM monitor
.....\$369.00

SB180-1-20
same as above w/ZCPR3, ZRDOS
and BIOS source.....\$499.00

-Quantity discounts available-

NEW

COMM180-M-S
optional peripheral board adds
1200 bps modem and SCSI
hard disk interface.

**TO ORDER
CALL TOLL FREE
1-800-635-3355**

**TELEX
643331**

For technical assistance or
to request a data sheet, call:

1-203-871-6170



Micromint, Inc.
25 Terrace Drive
Vernon, CT 06066

C CHEST

Listing One (Listing continued, text begins on page 22.)

```
103     for( i = 56 ; --i >= 0; putc( 0xc4 , stderr )
104     ;
105
106     E("\275");
107 }
108
109 /*-----*/
110
111 usage()
112 {
113     E("more: Copyright (C) 1986, Allen I. Holub. All rights reserved.");
114     E("\nUsage: more [+<num>] [file...] \n");
115     E("Print all files in list on the screen, pausing every 23 lines");
116     E("If + is specified, more will start printing at character <num>");
117     E("One of the following commands is executed after each page:");
118
119     help();
120     exit(1);
121 }
122
123 /*-----*/
124
125 push( file_posn )
126 long   file_posn;          /* Push file_posn onto the stack */
127 {
128     if( STACKFULL )        /* If the stack is full, compress it */
129         comp_stk();
130
131     *( --Sp ) = file_posn;
132 }
133
134 /*-----*/
135
136 long   pop()
137 {
138     /*      Pop one entry off the stack and return the file
139     *      position.
140     */
141
142     return STACKEMPTY ? 0 : *Sp++ ;
143 }
144
145 /*-----*/
146
147 comp_stk()
148 {
149     /*      Compress the stack by removing every other entry.
150     *      This routine is called when the stack is full (we've
151     *      read more lines than the stack can hold).
152     */
153
154     register STACKTYPE      *dest, *src;
155
156     fprintf(stderr, "\007Stack Full: Compressing\n");
157
158     src = dest = Stack + STACKSIZE;
159
160     while( (src -- 2) >= Stack )
161         *--dest = *src;
162
163     Sp = dest;
164 }
165
166 /*-----*/
167
168 getcon()
169 {
170     /*      Get one character from the console using a direct
171     *      bios call. Map \r into \n if one is encountered.
172     */
173
174     register int      c;
175
176     c = b_getc() & 0x7f;    /* Get a character from console */
177     putchar(c);            /* Echo character */
178
179     return ( c == '\r' ) ? '\n' : c ;
180 }
181
182 /*-----*/
183
184 clear_io()
185 {
186     /*      Clears the entire I/O queue, both at the bios and the
187     *      bdos level.
188     */
189
190     while( look() )
191         b_getc();
192
193     #ifdef NEVER
194         while( kbhit() )
195             getchar();
196     #endif
197 }
198
199 /*-----*/
200
201
202 khit()
203 {
204     if( look() )
205         /* Return true if a key has been hit on */
206         /* the physical keyboard (as compared */
207         /* with a character available on stdin) */
```

(continued on page 70)

PAINLESS WINDOWS.

Windows. Data Entry. Menus.
Finally, a C programmers' tool that makes
them as easy to use as *printf()*.
With Greenleaf DataWindowsTM,
you move in quantum leaps!

Snaazzy Window Treatments

DataWindows represents an important breakthrough in C programming tools. It sets you free so you can create exciting programs quickly and easily, saving both time and money! Developed to work with the IBM PC, XT, AT, compatibles, and MSDOS or PCDOS, DataWindows is a carefully tooled system of C functions which will jazz up your programs with unprecedented efficiency.

Greenleaf DataWindows is integrated windows, transaction data entry, pop-up, pull-down, and Lotus style menu systems with:

- **Screen Management.** You don't have to remember what's on the display or the sequence in which you put it there. DataWindows does the grunt work. There are no restrictions.
- **Transaction Data Entry.** Data entry windows can have any number of fields with sophisticated options for reading many data types. Calls are made to help, validation, and other functions. Full featured text editing, protected and mandatory fields, dBASE type picture strings, context sensitive help, validation of fields and transactions, redefinable keys, password entry, attribute control, keyboard idle and much more.
- **Device Independence.** It detects the type of display adapter your computer is using and adjusts to it automatically for CGA, EGA, or monochrome. Logical video attributes are easy to use for color or monochrome.
- **Compatibility.** Runs with Microsoft Windows and IBM TopView.
- **The Greenleaf Tradition of Quality.** Reliable products. Professional documentation that gets you up and running quickly and keeps you there. Reference card. Newsletter and Bulletin board.

IBM, Microsoft & dBase, are registered trademarks of International Business Machines, Microsoft Corporation & Ashton-Tate respectively. PCDOS, IBM PC, XT, AT, & TopView are trademarks of IBM; MSDOS and Microsoft Windows are trademarks of Microsoft Corporation.



Stop Window Shopping

Order Today. Or call toll free for a free demo of the windows library that makes all the others obsolete.

Order any of these high performance tools by calling your dealer or 1-800-523-9830 today. Specify compiler when ordering. Add \$8 for UPS second day air, or \$5 for ground. Texas residents add sales tax. MasterCard, VISA, P.O., check, COD. In stock, shipped next day.

Greenleaf DataWindows	\$225
DataWindows Source Module	\$225
The Greenleaf Comm Library v2.0	\$185
The Greenleaf Functions v3.0	\$185
Digiboard Comm/4-II	\$325
Digiboard Comm/8-II	\$535



GREENLEAF

Software®

1411 LeMay Drive, Suite 101
Carrollton, TX 75007

Call Toll Free

1-800-523-9830

In Texas and Alaska, call

214-446-8641

Circle no. 97 on reader service card.

Window Dressings

- **Simple or Complex Windows.** Up to 254 powerful overlaid windows simultaneously, all with just one kind of window to remember! Yet any window can be from one character to 32K!
- **Easy Window Operations.** DataWindows lets you move, zoom, frame, title, change colors, titles, frames, size, location, and make windows visible or invisible at will! Functions set cursor, attributes, and write data to any window or "current window". Word wrap, auto scroll, keyboard functions.
- **Write to Any Window Any Time.** Windows may be visible, overlaid, or invisible, and you can write to them anyway. What you write will be seen when the windows become visible.
- **DataWindows is fast!** It writes directly to video memory (in some modes).
- **Easy to save!** Any window, complete with attributes, can be saved on disk quickly and efficiently.
- **Source code available. No royalties.**

Also from Greenleaf:

The Greenleaf Functions v3.0

The most complete, mature C language function library for the IBM PC, XT, AT and close compatibles. Includes over 225 functions — DOS, disk, video, color text and graphics, string, time/date, keyboard, disk status and Ctrl-Break functions plus many more.

The Greenleaf Comm Library

Our 2.0 version is the hottest communications facility of its kind. Over 120 functions — ring buffered, interrupt driven asynchronous communications for up to 16 ports simultaneously with XMODEM, XON/XOFF, many many sophisticated features.

We support all popular C compilers for MSDOS/PCDOS: Microsoft, Lattice, Computer Innovations, Aztec, DeSmet, and others.



OUR NEW LOOK
To serve you better, we've expanded our advertised price list to provide a more complete selection of the high quality products we carry. And we've included more information about our policies and services.

apl language

APL*PLUS/PC by STSC	595	449
APL*PLUS/PC Spreadsheet Manager by STSC	195	159
APL*PLUS/PC Tools Vol 1 by STSC	295	239
APL*PLUS/PC Tools Vol 2 by STSC	150	129
APL*PLUS/UNIX For AT XENIX by STSC	995	795
Btrieve ISAM File Mgr by SoftCraft	250	194
Financial/Statistical Library by STSC	275	219
Pocket APL by STSC	95	79
STATGRAPHICS by STSC	795	619

arity products

Arity Expert System Development Pkg	295	279
Arity File Interchange Toolkit	50	48
Arity PROLOG Compiler & Interpreter	795	739
Arity PROLOG Interpreter	350	329
Arity Screen Design Toolkit	50	48
Arity SQL Development Package	295	279
Arity Standard Prolog	95	89

artificial intelligence

ESP ADVISOR by Expert Systems Intl	New	895	839
PROLOG-2 Interface	New	395	369
ExpertEDGE by Human Edge		795	659
Expertech II by IntelligenceWare		475	CALL
EXSYS Development Software by EXSYS		395	339
First Class by Human Edge		500	419
GCLISP Golden Common LISP by Gold Hill	New	495	CALL
GCLISP 286 Developer by Gold Hill		1190	CALL
Insight 1 by Level Five Research		95	75
Insight 2m by Level Five Research		485	389
Intelligence/Compiler IntelligenceWare		990	CALL
Logic-Line Series 1 by Thunderstone		90	85
Logic-Line Series 2 by Thunderstone		125	115
Logic-Line Series 3 by Thunderstone		150	139
LPA microPROLOG by Pgm Logic Systems		250	219
with APES		450	399
LPA Professional microPROLOG		395	349
with APES		695	599
Microsoft LISP Common LISP		250	189
PROLOG-2 Interpreter by ESI	New	450	419
PROLOG-2 Interpreter and Compiler	New	895	839
QNAL by NIAL Systems		375	359
Turbo PROLOG Compiler by Borland Intl		100	79

assembly language

386 ASM/LINK Cross Asm by Phar Lap	New	495	CALL
8088 Assembler by 2500 AD		100	89
ASMLIB —Function Library by BC Assoc		149	129
Cross Assemblers Various 2500 AD	CALL	CALL	CALL
Microsoft Macro Assembler		150	99
Visible Computer: 8088 Software Masters		80	65

basic language

BetterBASIC by Summit Software		200	164
8087 Math Support		99	84
Btrieve Interface		99	84
C Interface	CALL	CALL	CALL
Run-time Module		250	224
Microsoft QuickBASIC	New Version	99	79
Professional BASIC by Morgan		99	78
8087 Math Support		50	46
True Basic with BASICA Converter		200	109
True Basic w/Converter & Run-time	New	350	199
Advanced String Library	New	50	45
Asynch Communication Support		50	45
BASICA Converter		50	45
Btrieve Interface		50	45
Developer's Toolkit		50	45
Formlib		50	45
Hercules Graphic Support	New	50	45
Run-time Module		150	109
Sorting & Searching	New	50	45

blaise products

ASYNCH MANAGER Specify C or Pascal		175	136
C TOOLS		125	105
C TOOLS 2		100	84
C TOOLS PLUS	New	175	139
EXEC Program Chainer		95	79
PASCAL TOOLS		125	105
PASCAL TOOLS 2		100	84
PASCAL TOOLS & PASCAL TOOLS 2		175	139
RUNOFF Text Formatter	New	50	47
TURBO ASYNCH PLUS	New Version	100	84
TURBO POWER TOOLS PLUS	New Version	100	84
VIEW MANAGER Specify C or Pascal		275	208

borland products

REFLEX Data Base System		150	119
REFLEX Workshop		70	59
REFLEX & REFLEX Workshop		200	159
Turbo DATABASE TOOLBOX		70	54
Turbo EDITOR TOOLBOX		70	54
Turbo GAMEWORKS TOOLBOX		70	54
Turbo GRAPHIX TOOLBOX		70	54
Turbo LIGHTNING		99	75
Turbo PASCAL with 8087 and BCD		100	69
Turbo Prolog Compiler		100	79
Turbo TUTOR for Turbo PASCAL		40	32
Word Wizard	New	70	59
Word Wizard and Turbo Lightning	New	150	129

c compilers

C-86 by Computer Innovations		395	288
Datalight C Compiler Small Model		60	49
Datalight Developer Kit w/Large Model		99	79
DeSmet C w/Debugger		159	145
DeSmet C w/Debugger & Large Case		209	193
Eco-C Development System by Ecosoft		125	89
Lattice C Compiler from Lattice		500	299
Mark Williams Let's C		75	59
with csd Source Debugger		150	118
Mark Williams MWC-86		495	299
Microsoft C with CodeView	New Version	450	299
Wizard C Compiler by Wizard Systems		450	369

c interpreters

C-terp by Gimpel, Specify compiler		300	239
C Trainer by Catalystix	New	99	CALL
Instant C by Rational Systems		500	CALL
Introducing C by Computer Innovations		125	105
Run/C from Lifeboat		150	99
Run/C Professional from Lifeboat		250	184

c utilities

Also refer to Blaise, GSS, Lattice, Microsoft, Phoenix, Polytron, SoftCraft and XENIX sections.			
APT by Shaw American Technology		395	339
Basic C Library by C Source		175	135
C Essentials by Essential Software		100	85
C-ISAM by Relational Database Sys	New	225	199
C to dBase by Computer Innovations		150	139
C-tree ISAM File Manager by FairCom		395	329
C Utility Library by Essential Software		185	138
C Windows by Syscom		100	89
C Wings by Syscom		50	45
CI Probe by Computer Innovations		225	199
CI ROMPac by Computer Innovations		195	149
dbQUERY by Raima	New	CALL	CALL
dbVISTA Single-User DBMS by Raima		195	159
with Source Code		495	429
dbVISTA Multi-User DBMS by Raima		495	429
with Source Code		990	849
dBx dBase/C Translator by Desktop AI		350	329
Essential Graphics by Essential Software		250	209
Flash-up Windows by Software Bottling		75	69
GraphiC Mono v2.2 by Sci Endeavors		280	219
GraphiC Color v3.0 by Sci Endeavors		350	299
GRAFLIB by The Librarian	New	175	CALL
Greenleaf Functions by Greenleaf		185	135
Greenleaf Comm Library by Greenleaf		185	135
THE HAMMER by CES Systems		195	175
HELP/Control by MDS	New	125	109
MetaWINDOWS by Metagraphics		185	139
MetaWINDOWS/Plus by Metagraphics		235	199
Multi-Halo by Media Cybernetics		300	218
On-line Help from Opt-Tech Data Proc		149	119
PANEL by Roundhill Computer Systems		295	229

PC Lint by Gimpel Software	New Version	139	109
PLOTHI by The Librarian	New	175	CALL
PLOTHP by The Librarian	New	175	CALL
Sci Subroutine Library by Peerless		175	139
Vitamin C by Creative Programming		150	139
VC Screen Forms Designer		100	85
Zview by Data Management Consultants		245	198

cobol language

Micro Focus COBOL Workbench		4000	3479
Micro Focus Level II COBOL		1500	CALL
COGRAPHICS	New	250	219
COMATH		200	169
FORMS-2		300	269
Level II Animator		900	CALL
Level II SOURCEWRITER		2000	CALL
Micro Focus Level II COBOL for Novell		2000	1799
Micro Focus Micro/SPF		175	159
Micro Focus Professional COBOL		3000	2395
Multi-user Runtime for PC Network		500	449
Microsoft COBOL		700	495
Microsoft COBOL Tools w/Debugger		350	209
Realia COBOL		995	839
RM/COBOL by Ryan-McFarland		950	875
RM/COBOL 8X ANSI 85 by Ryan-McFarland		1250	695

debuggers & profilers

386 DEBUG Cross Debugger by Phar Lap	New	175	139
Advanced Trace-86 by Morgan Computing		175	139
CI Probe by Computer Innovations		225	199
Codesifter Profiler by David Smith		119	99
Codesmith-86 by Visual Age		145	108
DSD86 by Soft Advances	New	70	65
DSD87 by Soft Advances	New	100	89
Periscope I by Data Base Decisions		295	249
Periscope II w/NMI Breakout Switch		145	115
Periscope II-X Software only		115	95
The PROFILER with Source Code by DWB		125	95
The WATCHER Profiler by Stony Brook	New	60	55

forth language

CFORTH Native Code Compiler by LMI		300	239
Forth/83 Metacompiler Specify Target		750	599
PC/Forth by Laboratory Microsystems		150	119
PC/Forth+ by Laboratory Microsystems		250	209
Advanced Color Graphics Support		100	79
Enhanced Graphics Support		200	159
Intel 8087 Support		100	79
Interactive Symbolic Debugger		100	79
Native Code Optimizer		200	159
PCTERM Modern Pgm for Smartmodem		100	79
Software Floating Point		100	79

fortran language

50 MORE: FORTRAN by Peerless Engr		125	99
ACS Time Series by Alpha Computer		495	429
Btrieve ISAM File Mgr by SoftCraft		250	194
For-Winds by Alpha Computer Service		90	79
Fortlib-Plus by Alpha Computer Service		70	55
FORTLIB by The Librarian	New	95	CALL
FORTLIB Addenda by Impulse Engr	New	95	89
FORTLIB Addendum by Impulse Engr	New	165	149
GRAFLIB by The Librarian	New	175	CALL
I/O PRO with NO LIMIT Library			

by MEF	New	390	CALL
Microcompatibles Combo Package		240	219
Grafmatic		135	119
Plotmatic		135	119
Microsoft FORTRAN		350	215
Multi-Halo by Media Cybernetics		300	218
NO LIMIT by MEF Environmental	New	129	CALL
PANEL Screen Designer by Roundhill		295	229
PLOTHI by The Librarian	New	175	CALL
PLOTHP by The Librarian	New	175	CALL
RM/FORTLIB by Ryan-McFarland		595	394
Sci Subroutine Library by Peerless		175	139
Sci Subroutine Package by Alpha		295	259
Strings & Things by Alpha Computer		70	55

gss products

GSS Graphics Development Toolkit	New	495	379
GSS Kernel System for DOS	New	495	379
GSS Kernel System for IBM RT	New	795	649
GSS Metafile Interpreter	New	295	239
GSS Plotting System	New	495	379
GSS SOLUTIONS Chart	New	295	239
GSS SOLUTIONS Plottalk	New	295	239
GSS SOLUTIONS Terminal	New	295	239

lattice products

Lattice C Compiler from Lattice		500	299
with Library Source Code		900	549
C Cross Reference Generator		50	39
with Source Code		200	159
C-Food Smorgasbord Function Library		150	99
with Source Code		300	195
C-Sprite Source Level Debugger		175	138
Curses Screen Manager		125	98
with Source Code		250	194
dBx dBase File Manager for C		250	194
with Source Code		500	388

Your Quality Connection .

FREE SHIPPING

Orders within the USA are shipped FREE via standard UPS. Express shipping is available at the shipping carrier's standard rate with no handling charge.

CREDIT CARDS

VISA and MasterCard are accepted at no extra cost. Your card is charged when your order is shipped. Mail orders please include credit card expiration date.

COD's AND PO's

COD's and Purchase Orders are accepted at no extra cost. PO's with net 30-day terms are available to qualified US accounts

FOREIGN ORDERS

Foreign and Canadian mail orders please include sufficient funds for shipping (excess payment refunded with your order). Foreign orders (except Canada), please include \$5 for customs form preparation. All transactions are in US dollars.

VOLUME ORDERS

Call for special pricing.

SOUND ADVICE

Our knowledgeable technical staff can compare products, answer technical questions and send you detailed product information tailored to your needs.

30 DAY GUARANTEE

Most of our products come with a 30 day evaluation period or a 30 day return guarantee. Please note that some products are restricted by their manufacturers from this guarantee. Call for more information.

CALL TOLL FREE

US 800-336-1166

CANADA 800-225-1166

OHIO 216-877-3781

CUSTOMER SERVICE 216-877-1110

Hours: Weekdays 8:30 AM to 8:00 PM EST.

Ohio customers please add 5% state sales tax.

Call or write for our FREE comprehensive price guide.

LMK Make Facility	195	149
RPG II Compiler No Royalties	750	635
SecretDisk File Encryption Utility	120	95
SideTalk Resident Communications	120	95
Text Management Utilities	120	95
TopView Toolbasket Function Library	250	194
with Source Code	500	388
Z-80 C Cross Compiler	500	388
with Library Source Code	1000	789

logitech products

LOGIMOUSE C7 Mouse Hardware	New	99	89
with PLUS Pkg	New	119	105
with PLUS Pkg & PC Paintbrush	New	169	149
with PLUS Pkg & CAD Software	New	189	165
MODULA-2/86 Compiler		89	64
MODULA-2/86 with 8087 Support		129	104
MODULA-2/86 PLUS		189	148
Library Sources		99	89
Make Utility	New	29	27
ROM Package	New	199	179
Run Time Debugger		69	59
Turbo to Modula Translator	New	49	45
Utilities Package		49	45
Window Package	New	49	45

microsoft products

Microsoft BASIC Interpreter for XENIX	350	279
Microsoft C with CodeView	450	299
Microsoft COBOL Compiler	700	495
for XENIX	995	789
Microsoft COBOL Tools with Debugger	350	209
for XENIX	450	329
Microsoft FORTRAN Compiler	350	215
for XENIX	695	589
Microsoft LISP Common LISP	250	185
Microsoft Macro Assembler	150	99
Microsoft Mouse Bus Version	175	149
Microsoft Mouse Serial Version	195	159
Microsoft muMath Includes muSIMP	300	195
Microsoft Pascal Compiler	300	195
for XENIX	695	589
Microsoft QuickBASIC	New version	99 79
Microsoft Sort		195 149
Microsoft Tech Ref Encyclopedia	New	135 CALL
Microsoft Windows		99 74
Microsoft Windows Development Kit		500 395

other languages

CCS MUMPS Single-User by MGlobal	60	55
CCS MUMPS Multi-User by MGlobal	450	379
Janus/ADA C Pack by R&R Software	95	89
Janus/ADA D Pack by R&R Software	900	795
Methods Smalltalk by Digital	79	69
Smalltalk/V by Digital	99	89
SNOBOL4+ by CatSpaw	95	84

other products

Dan Bricklin's Demo Pgm Software Garden		
FASTBACK by 5th Generation Systems		
Informix for DOS by RDS	New	
Informix4GL for DOS by RDS	New	
InformixSQL for DOS by RDS	New	
Inside Track from Micro Help	New	
Interactive EASYFLOW by Haventree		
MACH 2 by Micro Help	New	
MKS Toolkit with vi Editor by MKS	New	
OPT-Tech Sort by Opt-Tech Data Proc		
Peeks 'n Pokes from Micro Help	New	
PrintQ by Software Directions	New	
Quilt Computing Combo Package	New	
QMake Program Rebuild Utility	New	
SRMS Software Revision Mgmt Sys		
Source Print by Aldebaran Labs		
Taskview by Sunny Hill Software	New	

phoenix products

Pasm86 Macro Assembler	New Version	195	CALL
Pdisk Hard Disk Utility	New	195	CALL
Plantasy Pac		1295	889
Plinix Performance Analyzer	New Version	395	249
Plix-86 Program Debugger		195	139
Plix-86 Plus Symbolic Debugger		395	249
PforCe Comprehensive C Library		395	289
Plink-86 Overlay Linker		395	249
Plink-86 Plus Enhanced Linker		495	355
Pmaker Make Utility	New Version	125	CALL
Pmate Macro Text Editor		195	145
Pre-C Lint Utility	New Version	295	CALL
Ptel Binary File Transfer Program		195	145

PolyMake UNIX-like Make Facility	99	79
PolyOverlay Overlay Optimizer	99	79
PolyWindows Products All Varieties	CALL	CALL
PolyXREF Complete Cross Ref Utility	219	179
PolyXREF One language only	129	109
PVCS Version Control System	395	329
PVMFM Virtual Memory File Manager	199	149

softcraft products

Btrieve ISAM Mgr with No Royalties	250	194
Xtrieve Btrieve Query Utility	195	165
Rtrieve Xtrieve Report Generator	85	75
Btrieve/N for Networks	595	464
Xtrieve/N for Btrieve/N	395	299
Rtrieve/N for Xtrieve/N	175	159

text editors

Brief from Solution Systems	195	CALL
Epsilon Emacs-like editor by Lugaru	195	165
FirstTime for Turbo by Spruce Tech	75	69
KEDIT by Mansfield Software	125	109
PC/VI by Custom Software Systems	149	129
Personal REXX by Mansfield Software <i>New</i>	125	109
SPF/PC by Command Technology Corp	195	165
Vedit by CompuView	150	115
Vedit Plus by CompuView	225	179
XTC with Pascal source by Wenden	99	84

turbo pascal utilities

See also Blaise, Borland and SoftCraft sections.

ALICE Interpreter by Software Channels	95	69
Btrieve ISAM File Mgr See SoftCraft	250	194
FirstTime for Turbo by Spruce Tech	75	69
Flash-up Windows by Software Bottling	75	69
HELP/Control by MDS New	125	109
On-line Help from Opt-Tech Data Proc	149	119
Screen Sculptor by Software Bottling	125	95
T-Debug by TurboPower Software . . . New	60	55
Turbo EXTENDER by TurboPower Software . .	85	69
Turbo Professional by Sunny Hill	70	49
TurboHALO from IMSI New	99	85
TurboPower Utilities by TurboPower Sftwr .	95	84
TurboWINDOW by MetaGraphics	80	69

wenden products

Operating System Toolbox	99	84
PCNX Operating system	99	84
PCVMS Similar to VAX/VMS	99	84
XTC Text editor with Pascal source	99	84

xenix system v

Complete XENIX System by SCO	1295	1099
XENIX Development System	595	529
XENIX Operating Sys Specify XT/AT	595	529
XENIX Text Processing Package	195	155

xenix products

APL+PLUS/UNIX For AT XENIX by STSC . . .	995	795
Btrieve ISAM File Mgr by SoftCraft	595	465
C-ISAM by Relational Database Sys . . . New	319	289
c-tree ISAM Mgr w/Source by FairCom . . .	395	329
dBx dBase/C Translator by Desktop AI . . .	550	499
dbVISTA Single User w/Source by Raima . .	495	429
dbVISTA Multi User by Raima	495	429
dbVISTA Multi User w/Source by Raima . . .	990	849
Informix by Relational Database Sys	995	839
Informix4GL by RDS	1500	1279
InformixSQL by RDS	995	839
Lyrix by SCO	595	489
Micro Focus Level II Compact COBOL	1000	899
Forms-2	400	359
Level II ANIMATOR	600	539
Microsoft BASIC Interpreter	350	279
Microsoft COBOL Compiler	995	789
Microsoft COBOL Tools —with Debugger . .	450	329
Microsoft FORTRAN Compiler	695	589
Microsoft Pascal Compiler	695	589
Networks for XENIX by SCO	595	529
PANEL Screen Designer by Roundhill	595	539
REAL-TOOLS C Library by PCT New	500	CALL
REAL-TOOLS w/Complete Source Code	5000	CALL
RM/COBOL by Ryan-McFarland	1250	995
RM/FORTRAN by Ryan-McFarland	750	599
SCO Professional Lotus clone by SCO	795	695

Prices are subject to change.

Programmer's Connection Incorporated
136 Sunnyside Street
Hartville, Ohio 44632

programmer's connection

Complete C Programs in Half the Time, with *Instant-C*TM

You can create programs much faster with Instant-C than with conventional programming tools. How? Because Instant-C is a high-performance interpreter; there are **no compile or link delays**. Change your program, then test it immediately. No matter how large your program, the turnaround time is just seconds.

"Instant-C means instant gratification."—*PC Magazine*, **Editor's Choice** for best C interpreter.

Powerful **source-level debugging** saves your time. Conditional breakpoints, single-stepping by statement, source code backtraces, data monitoring, and many other debugging features make it easy to wipe out bugs quickly. Direct execution of any statement or function makes testing a breeze.

"The resulting debugging and testing capabilities are fantastic and the detailed trace/debug/display commands make it easy."—*The C Journal*

Instant-C checks pointer references for reasonableness, and checks that array indexes are within declared bounds. This **run-time checking** stops your program as soon as errors occur, for easiest debugging.

Not only does Instant-C help you quickly change, test, check and debug your code, but it runs your program **fast enough for real-time** applications.

"It is much faster than any of the other products mentioned and was the only one able to complete the standard SIEVE in a reasonable time. Clearly, this high speed allows much more complex problems to be attacked with Instant-C than with any of the other products discussed."—*Computer Language*

Immediate feedback and precise diagnostics make Instant-C great for learning C. Full K&R and the ability to **link compiled object code and libraries** (Lattice and Microsoft) makes Instant-C compatible with your existing programs.

Instant-C makes all parts of the programming task as fast as possible.

"Clearly, Instant-C is the performance champion."—*PC Tech Journal*

Version 2 works with MS-DOS and PC-DOS, and has a full 31 day **money back guarantee**. Instant-C is only \$495. Order today! Call or write for full information.

Rational P.O. Box 480
Systems, Inc. Natick, MA 01760
(617) 653-6194

C CHEST

Listing One (Listing continued, text begins on page 22.)

```

206         clear_io();
207         return 1;
208     }
209     return 0;
210 }
211
212 /*-----*/
213
214 char *getbuf( p )
215 register char *p;
216 {
217     /* Get a line of input using direct console I/O and put it
218      * into buf. Return a pointer to the first whitespace on the
219      * line or to end of line if none. This routine is for
220      * getting commands from the user, not for getting normal
221      * input. ^H is supported as a destructive backspace but no
222      * other editing is available.
223      */
224
225     register int c;
226     int gottail = 0;
227     char *start = p;
228     char *tail = "";
229
230     clear_io();
231
232     while( (c=getcon()) != '\n' )
233     {
234         if( c == '\b' )
235         {
236             if( p <= start )
237                 fputs( " \007", stderr );
238             else
239             {
240                 --p;
241                 fputs( " \b", stderr );
242             }
243         }
244         else
245         {
246             if( isspace(c) && !gottail )
247                 gottail = (int)( tail = p );
248             *p++ = c;
249         }
250     }
251
252     *p = '\0';
253     return( p <= start ? NULL : tail );
254 }
255
256 /*-----*/
257
258 percent(s)
259 char *s;
260 {
261     /* Print the percentage of the file that we've seen so far */
262
263     printf("%4.1f%%s", ((double)TOS / (double)Flen) * 100.00, s );
264 }
265
266 /*-----*/
267
268 int getcmd()
269 {
270     /* Get a command from the keyboard, using direct
271      * bios I/O. Commands take the form [num]<C>. Returns
272      * the command. Repeat_count is initialized to hold [num]
273      * or 1 if no num is entered.
274      */
275
276     int c;
277
278     clear_io();
279     percent("");
280     printf(" , line %ld ( ? for commands): ", Line );
281
282     Repeat_count = 0;
283     while( '0' <= (c = getcon()) && c <= '9' )
284         Repeat_count = (Repeat_count * 10) + (c - '0');
285
286     if( Repeat_count == 0 )
287         Repeat_count = 1;
288
289     erase_line();
290
291     if( c == 0x03 ) /* ^C == abort */
292         exit( 1 );
293
294     return( c );
295 }
296
297 /*-----*/
298
299 char *inputline( suppress )
300 {
301     /* Get a line from the file being processed and put it into
302      * buf. Push the start of line character onto the stack.
303      * return 0 on end of file, a pointer to the line (ie. to buf)
304      * otherwise.
305      */
306
307     register int rval;
308     register long start_of_line;
309     static char buf[BSIZE];

```

```

310 start_of_line = ftell( Ifile );
311
312 if( rval = (int) fgets(buf, BSIZE, Ifile) )
313 {
314     line++;
315     push( start_of_line );
316     if( !suppress )
317         fputs( buf, stdout );
318 }
319
320 return rval ? buf : NULL ;
321
322 }
323
324 /*-----*/
325
326 printpage()
327 {
328     /* Print an entire page from the input file */
329
330     register int i;
331
332     for( i = PAGESIZE-1; --i >= 0 && inputline(0); )
333         ;
334 }
335

```

(continued on next page)



LPI-PL/I lets you move your PL/I applications to today's hottest new workstations without going through the agony of re-programming.



LPI-RPGII, LPI-COBOL, LPI-PL/I, LPI-BASIC, LPI-FORTRAN, LPI-PASCAL, LPI-C, LPI-DEBUG
LANGUAGE PROCESSORS, INC., 400-1 TOTTEN POND RD., WALTHAM, MA 02154 (617) 890-1155

Circle no. 266 on reader service card.

WARP SPEED LIGHT PEN

RUN AUTOCAD AS FAST AS YOU THINK

Light Pens, the natural graphic input device, are finally available for AUTOCAD. Light Pens, Software and single pixel, high resolution convertor cards are ready for immediate delivery.

For Information Call: In Calif. (800) 826-1563

(800) 874-4315



WARP SPEED
COMPUTER PRODUCTS INC.
5555 S. Inglewood Boulevard
Los Angeles, California 90230



Circle no. 189 on reader service card.



FAST

MICROSOFT
QuickBASIC
Compiler
New Version 2.0

List Price \$99 Our Price \$79

Performance beyond BASICA.

- Quickly compiles BASICA Interpreter programs into native code.
- Can compile entirely in memory for even faster compilation. New!
- Programs execute up to ten times faster.
- Can call high speed assembly routines.
- Network file sharing with record locking.
- Graphics (incl. EGA), sound and music.
- Numeric arrays, each up to 64K bytes, can use up to available memory. New!
- Programs can be distributed without source and runtime fees.

Complete development environment.

- Built-in editor and debugger to write programs and locate and fix errors. New!
- Execution tracing at source level. New!
- Menu-driven interface that supports either keyboard or optional mouse. New!

Structured & modular programs.

- Update and maintain programs effortlessly.
- Multi-line IF/THEN/ELSE statements. New!
- Alphanumeric labels — line labels are optional — for easy to read programs.
- Subprograms that use local and global variables.
- Programs use up to available memory by compiling and linking individual modules.
- Includes a library of routines to access DOS and BIOS interrupts. New!

Send one dollar and we'll send you a Microsoft QuickBASIC demonstration diskette.

US 800-336-1166

CANADA 800-225-1166

Ohio & Overseas 216-877-3781

programmer's connection

136 SUNNYSIDE ST. HARTVILLE, OHIO 44632

Circle no. 98 on reader service card.

Listing One (Listing continued, text begins on page 22.)

```

336 /*-----*/
337
338 search()
339 {
340     /* Prompt for a pattern and then search for it in the
341      * file. Stop searching if the pattern is found or if
342      * any key is hit. The previous pattern is remembered
343      * in a local array so, if CR is entered instead of a
344      * pattern, the previous pattern is used.
345      */
346
347     static char pat[128], opat[128];
348     char *iline;
349     extern int *makepat();
350     int *template;
351
352     printf("/");
353
354     if( !getbuf( pat ) )
355         strcpy( pat, opat );
356
357     if( !(template = makepat( pat, 0 )) )
358         printf("Illegal regular expression: %s\n", pat );
359     else
360     {
361         erase_line();
362         printf("/%s\n", pat );
363
364         while( (iline = inputline(1)) && !khit() )
365         {
366             percent("\r");
367             if( matches( iline, template, 0 ) )
368                 break;
369         }
370
371         unmakepat( template );
372         fseek( Ifile, pop(), 0 ); /* back up one line */
373         --line;
374         line ( 0xcd, 1);
375         printpage();
376     }
377     strcpy( opat, pat );
378

```

C Bricklin Run

A Primer for Modern Times, or How to Use **C-scape** to Turn Dan Bricklin's Demo Screens into C Code:

Jane writes software. Dick is Jane's client. Dick wants his software yesterday. Jane uses Dan Bricklin's Demo Program to prototype Dick's software. That makes Dick and Jane happy.

Now Dick wants the real thing. Jane uses **C-scape's demo2c** utility to turn Bricklin's screens into C code with text, menus, input fields, and colors. Next Jane links with **C-scape's library** to add scrolling and type support.

Jane's program runs. Dick changes specs. Jane fixes the code easily with **C-scape**. See Dick smile. Dick is one happy client.

Look at **C-scape**. C Bricklin run. See productivity soar.

That's right. Compilable C code created directly from each and every screen you produce using Dan Bricklin's Demo Program. And the procedure is as simple as Dick and Jane. Make **C-scape** your primer.


- | | |
|---|--|
| <input type="checkbox"/> scrolling | <input type="checkbox"/> colors |
| <input type="checkbox"/> full type support | <input type="checkbox"/> completely definable keys |
| <input type="checkbox"/> fully definable validation | <input type="checkbox"/> full screen I/O support |
| <input type="checkbox"/> fields, menus, prompts, and text | <input type="checkbox"/> flexible, easy to learn and use |
| <input type="checkbox"/> Lotus-like and pull-down menus | <input type="checkbox"/> source code included |

30-day guarantee: Try C-scape for 30 days and see how it simplifies your development process. Following registration you'll get full source code and support. No run-time license, no royalties. We're developers, too. We know your needs.

\$149.00 C-scape (Lattice 3.0/Microsoft 3.0; others call)

\$219.00 C-scape with Dan Bricklin's Demo Program

Please add \$3.00 for shipping. Massachusetts orders please include 5% sales tax.

Another Key to Freedom, from
Oakland Group, Inc. 
675 Massachusetts Avenue, Cambridge, MA 02139



For orders and information, call:
617-491-7311 or
800-233-3733

```

379 )
380
381 /*-----*/
382
383 execute()
384 {
385     /*      Spawn off a child process. When the process terminates
386     *      print a message and redraw the current page. Note that
387     *      spawn() is used (rather than system()) so you can't
388     *      execute a batch file or a built-in command. This
389     *      subroutine will set the CMDLINE environment variable
390     *      to a null string for the sake of those routines that
391     *      are executing under the shell which will use it.
392     */
393
394     static char   buf[128];
395     char          *tail = " ";
396
397     static char   obuf[128], *otail = obuf;
398
399     register char *p;
400     register int  c;
401     int          got_tail = 0;
402
403     printf("!\n");
404
405     if( !(tail = getbuf(buf)) )          /* If no command entered, */
406     {                                   /* use the same one we      */
407         tail = otail;                   /* used last time         */
408         memcpy( buf, obuf, 128 );
409         printf( "\n!%s %s\n", buf, tail );
410     }
411     else
412     {
413         if( *tail )
414             *tail++ = '\0';
415     }
416
417     if( HAS_DOT(buf) )
418     {
419         /* Spawnlp will actually try to execute any file that you
420         * give it. If you say to execute an ASCII file, it will
421         * load that file into memory, try to execute it, and die
422         * a horrible death. We attempt to avoid this by checking
423         * for a dot in the file name. You may want to put a
424         * more rigorous test here.
425         */
426
427         fprintf(stderr, "\007<%s> is not a command\n", buf);
428     }
429     else
430     {
431         putenv("CMDLINE=");
432         if( spawnlp(P_WAIT, buf, buf, tail, NULL) == -1)
433             fprintf(stderr, "Can't execute <%s>\n", buf, tail );
434     }
435
436     printf("Hit any key to continue ....");
437     getcon();
438     erase_line();
439     putchar('\n');
440
441     otail = tail;
442     memcpy( obuf, buf, 128 );
443 }
444
445 /*-----*/
446
447 line( c , newline )
448 {
449     /*      Print a line of characters to mark top of page. 0xcd
450     *      is the IBM graphics character for a horizontal double
451     *      line. The cursor is put at the beginning of next line
452     *      if "newline" is true, else it's put at beginning of
453     *      current line.
454     */
455
456     register int  i;
457
458     putchar('\r');
459
460     for( i = 79; --i >= 0 ; putchar( c ) )
461         ;
462
463     putchar( newline ? '\n' : '\r' );
464 }
465
466 /*-----*/
467
468 backpage( count )
469 {
470     /*      Go back count pages and print the resulting page.
471     */
472
473     register int  i;
474
475     i = ((count+1) * PAGE_SIZE) -1;
476
477     while( --i >= 0 )
478     {
479         Line--;
480         pop();
481     }
482
483     line( 0xcd, 1);
484     fseek( ifile, pop(), 0 );
485     Line = max( Line - 1, 0 );
486     printpage();
487 }

```

(continued on next page)



FAST

MICROSOFT
FORTRAN Compiler
 Version 3.31

MS-DOS List Price \$350 Our Price \$215

XENIX List Price \$695 Our Price \$589

Mainframe Power.

- Native code compiler implements most ANSI '77 features plus extensions.
- Links with Microsoft C, Pascal and MASM.
- Port mainframe or minicomputer programs with little or no modification.
- COMMON blocks and arrays >64K.
- Full set of math libraries:
 - 8087/80286 emulation.
 - 8087/80287 coprocessor support.
 - Floating Point without 8087/80287.
 - BCD Floating Point.
- Single- and double- precision complex numbers and functions.
- Wide support by third party libraries.
- MS-DOS and XENIX source compatibility.
- Network file sharing with record locking.
- Conditional compilation.
- Separate module compilations w/overlays.
- Character substrings.
- Source debugging with SYMDEB utility (available with MS Macro Assembler).

Plus powerful utilities.

- LINK links very large programs (over one megabyte) using overlays.
- LIB creates, organizes and maintains object module libraries.
- EXEPACK (MS-DOS only) packs EXE files for smaller size and faster loading.
- EXEMOD (MS-DOS only) lets you modify fields in EXE file headers.

US 800-336-1166
 CANADA 800-225-1166
 Ohio & Overseas 216-877-3781

programmer's connection

136 SUNNYSIDE ST. HARTVILLE, OHIO 44632

Circle no. 103 on reader service card.

C CHEST

Listing One (Listing continued, text begins on page 22.)

```

489 }
490
491 /*-----*/
492
493 docmd( cmd, ateof )
494 {
495     /* Do a single command, return 1 if next file is requested.
496     * Actually call exit on a "quit" command or ^C.
497     */
498
499     register int    rval = 0;
500     register int    i;
501     long            posn;
502
503     do {
504         switch( cmd )
505         {
506             case CAN:      break;          /* NOP          */
507             case 'q':      exit(0);        /* abort        */
508
509             case '\n':
510                 if( ateof )                /* FORWARD MOTION */
511                     rval = 1;              /* one line       */
512                 else
513                     inputline(0);
514                 break;
515
516             case ':':
517                 if( ateof )                /* one page      */
518                     rval = 1;
519                 printpage();
520                 break;
521
522             case 'e':
523                 /* To end of file          */
524                 erase_line();
525                 while( inputline(1) && !khit() )
526                     percent("\r");
527                 break;
528
529             case 's':
530                 /* one line w/o printing */
531                 if( ateof )
532                     rval = 1;
533                 else
534                 {
535                     erase_line();
536                     inputline(1);
537                     percent("\r");
538                 }
539                 break;
540
541             case ESC:
542                 /* scroll till key is hit */
543                 if( ateof )
544                     rval = 1;
545                 else
546                     while( inputline(0) && !khit() )
547                         clear_io();
548
549                 clear_io();
550                 Repeat_count = 0;          /* Ignore repeat count */
551                 break;                    /* if it's set          */
552         }
553     }
554 }

```

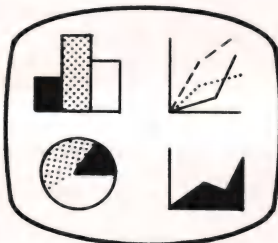
Tools For Programmers

ESSENTIAL GRAPHICS

\$250

C UTILITY LIBRARY

\$185



- Pascal • Fortran • C
- Fastest Library Available
- Powerful and Easy to Use
- Major Graphics Boards
- No Royalties

- 350 + C Functions, Source Included
- Pop-up Windows and Menus
- Fastest Screen Output Available
- Strings, Files, Keyboard, Mice
- No Royalties

C ESSENTIALS

\$49

**FALL
Special**

the 200 most frequently used functions



ESSENTIAL SOFTWARE, INC.

P.O. Box 1003 Maplewood, NJ 07040 914/762-6605

Circle no. 138 on reader service card.

```

551     case 'n':                /* to next file      */
552         rval = 1;
553         break;
554
555     case '/':                /* search for pattern */
556         search();
557         break;
558
559     case 'r':                /* to start of file   */
560         line( 0xcd, 1 );
561         CLEAR_STACK();
562         Line = 0;
563         fseek( Ifile, 0L, 0 );
564         printpage();
565         break;
566
567     case 'b':                /* to previous page   */
568         backpage( Repeat_count );
569         Repeat_count = 0;
570         break;
571
572     case 'o':                /* print file position */
573
574         printf("Top line = %ld, ",    BACK_SCRN );
575         printf("Bottom line = %ld\n",  TOS );
576         break;
577
578     case '!':
579         /* Close the file and spawn another shell.
580          * when we come back, reopen the file
581          * and position to the same place we
582          * were before. This is necessary because of
583          * a bug in Microsoft C ver. 3.0's spawn functions
584          * (they trash the IOB). It will cause problems
585          * if standard input is used as the input source
586          * (as in a pipe) because we won't be able to
587          * successfully reopen stdin.
588          */
589
590         Repeat_count = 0;        /* Ignore repeat count */
591         fclose( Ifile );
592         execute();
593         posn = pop();
594
595         if( Ifile = fopen(Ifile_name, "r") )
596         {
597             fseek( Ifile, posn, 0 );
598             backpage( 0 );
599         }
600         else
601         {
602             fprintf(stderr, "more: can't open %s\n",
603                          Ifile_name);
604             rval = 1;
605         }
606         break;
607
608     default :                /* Print the help msg. */
609         help();
610         cmd = getcmd();        /* get a new command */
611         Repeat_count++;
612         break;
613 }
614
615 ) while( --Repeat_count > 0 );
616
617 return( rval );
618 }
619
620 /*-----*/
621
622 dofile( fname )
623 char *fname;
624 {
625     /* Process lines from an input file having the indicated
626     * name.
627     */
628
629     if( (Ifile_name = fname) && !(Ifile = fopen(fname, "r")) )
630         fprintf(stderr, "more: can't open %s\n", fname );
631     else
632     {
633         Flen = filelength( fileno(Ifile) );
634         fseek( Ifile, Start_here, 0 );
635
636         CLEAR_STACK();
637         docmd(' ', 0 );        /* dump the first page */
638
639         for(;;)
640         {
641             for(;;)
642             {
643                 if( docmd( getcmd(), 0 ) )
644                     return;
645
646                 if( feof(Ifile) )
647                     break;
648             }
649
650             E("\n\020\020\020 LAST LINE IN FILE \021\021\021");
651             if( docmd( getcmd(), 1 ) )
652                 break;
653         }
654
655         fclose( Ifile );
656     }
657 }
658
659 /*-----*/
660

```

(continued on next page)



FAST

MICROSOFT
C Compiler
New Version 4.0
with CodeView Source Debugger

List Price \$450 Our Price \$299

The professional C compiler.

- Library routines implement most of UNIX System V and proposed ANSI C libraries.
- Implements register variables.
- Generates very fast and optimized code.
- Five memory models with new huge and compact.
- Mixed models using near, far and huge pointers.
- Support for huge programs up to one MB.
- Separate module compilation.
- IEEE single- and double-precision reals.
- Network file sharing with record locking.
- Support for pathnames and I/O redirection.
- Support for mixed language programming.
- Source and object compatible with XENIX.
- Support for huge arrays (>64K).
- Start up source helps create ROMable code.
- Microsoft Windows support.
- Includes LINK, LIB, MAKE, EXEPACK, EXEMOD.

CodeView source-level debugger.

- Multi-window display.
- Debug using source code, disassembly or both intermingled.
- Can display values of local and global variables and expressions as you debug.
- Set conditional breakpoints on variables or memory, trace and single step.
- Displays CPU registers and flags.
- Easily debug graphics-oriented programs.
- Accepts familiar SYMDEB or DEBUG commands.
- On-line help and drop-down menus.
- Keyboard or optional mouse support.

Send one dollar and we'll send you a Microsoft C/CodeView demonstration diskette.

US 800-336-1166
CANADA 800-225-1166
Ohio & Overseas 216-877-3781

programmer's connection

136 SUNNYSIDE ST. HARTVILLE, OHIO 44632

Circle no. 86 on reader service card.

Listing One (Listing continued, text begins on page 22.)

```

661 main(argc, argv)
662 char  **argv;
663 {
664     ctlc();
665     reargv(&argc, &argv);
666
667     if( argc > 1 )
668     {
669         if( argv[1][0] == '-' )
670             usage();
671
672         else if( argv[1][0] == '+' )
673         {
674             Start_here = atol( &argv[1][1] );
675             printf("Starting at character %ld\n", Start_here );
676             push ( Start_here );
677             ++argv;
678             --argc;
679         }
680     }
681
682     if( argc <= 1 )
683         dofile( NULL );
684     else
685         for( --argc > 0 ; dofile(++argv) )
686             ;
687
688     exit(0);
689 }

```

End Listing One

(Listing Two begins on page 78.)

FULL POWER DEBUGGING

DSD86, The PC-DOS Debugger 69.95
DSD87, The PC-DOS Debugger with 8087 Support . 99.95
DSD80, The CP/M Debugger 125.00

 **SoftAdvances**

P.O. Box 49473 • Austin, Texas 78765 • (512) 478-4763
1-800-232-8088

Circle no. 83 on reader service card.



Power Tools for system builders™

Call today for our free catalog of design aids, compilers, libraries, debuggers, and support tools for Apple and IBM micro computers. The Power Tools catalog includes product descriptions, warranty and license terms, and all the information you need to make an intelligent purchase decision.

TSF offers technical support, competitive pricing, free UPS shipping on orders over \$100, and a reasonable return policy. Visa, MasterCard, and American Express accepted without surcharge. **TSF helps you get your job done.**

Sample Prices:

MS Quick Basic \$49.00
Periscope II \$129.00
DS Backup \$64.95
10 Diskette Mailers \$4.95
Gimpel PC Lint \$139.00
PCMacBasic \$100.00

Call Toll Free
24 hrs a day/7 days a week

Ask For Operator 2053

800-543-6277

Calif: 800-368-7600



TSF

The Software Family™

• Dept. C-2 • 649 Mission Street
• San Francisco • CA 94105
• (415) 957-0111

Circle no. 230 on reader service card.

C DYNAMO™

FREE! UNIX[†] WORKALIKE FREE!

(See Limited Time Special Offer Below)

PAINT TEXT, HELP, MENU & DATA ENTRY SCREENS & WINDOWS FROM KEYBOARD • COMPLETE INPUT FORMATTING • FULL VALIDATION • UNLIMITED WINDOWS • AUTOMATIC WINDOW OVERLAY & RESTORE • FULL ATTRIBUTE CONTROL & HIGHLIGHTING • 1-2-3 LIKE MENUS • DRAW BOXES & BORDERS • GENERATE "C" CODE WITH TOUCH OF A KEY • LEFT/RIGHT/CENTER JUSTIFICATION • SPECIFY ANY FILL CHARACTER & CURRENCY SYMBOL • U.S./EUROPEAN NUMBER STYLES • MORE DATA TYPES • ONE CALL WILL OPERATE ENTIRE SYSTEM • SUPPORTS IBM PC/XT/AT & COMPATIBLES • SUPPORTS MICROSOFT, COMPILER LATICE, AZTEC, C186, ANY FULL K & R OR ANSI COMPILER

NEW
DYNAMO DATA ENTRY
UNIQUE, POWERFUL, NECESSARY
Full Validation of Each Keystroke
Range Checking
Data Security to Item Level
Scrollable Data Entry Forms with Fixed & Scrollable Parts
Allows Forms Larger Than Screen
Over 34 Item Types
Powerful "Picture" Capability
Unique: Mix Text, Data Entry Fields With Menu Items
Full source code. No Royalties
Code plus manual \$129.95

POWER WINDOWS

MOST POWERFUL YET POP-UP WINDOWS FOR

Menus/Overlays
Help Screens
Messages/Alarms

ZAP ON/OFF SCREEN FILE-WINDOW MANAGEMENT COMPLETE CONTROL OF:

Cursor
Attributes
Borders

AUTOMATIC

Horizontal & Vertical Scrolling
Word Wrap
Line Insertion

The most powerful, flexible and easy to use windowing package available! Many types of menus. Highlighting. Move data between files, keyboard, program and windows. Status lines. Change size/location/overlapping. Move/add/delete/cascade windows. Full source code. No royalties.
3 disks \$129.95

C FUNCTION LIBRARY

BEST YOU CAN GET

325 FUNCTIONS

SUPERB DOCUMENTATION

Most complete screen handling plus graphics/cursor/keyboard/data entry, 72 string functions with word wrap; status and control; utility/DOS BIOS/time/date functions; printer control & more. Special functions. Full source code. No royalties.
4 disks \$129.95

NEW DYNAMO SCREEN PAINTER AND FORMS CREATOR

DOES IT ALL

RIGHT FROM YOUR KEYBOARD AUTOMATIC CODE GENERATION

Data or Help Screens & Windows
Data Entry Screens & Windows
Menus

FAST, FLEXIBLE, EASY

Save Man-Months of Programming
Full Control of Screen Attributes
Monochrome or Color

FAST EASY MENU GENERATION

1-2-3 Like, Many Others
Full Read/Write Security by Item

Requires Dynamo Data Entry
Screen painter & manual \$129.95

B-TREE LIBRARY & ISAM DRIVER

POWERFUL DATA MANAGER

Fixed/Variable length records
Fast! Easy To Use!
16.7 Million Records/File
16.7 Million Keys/File

Full source. No royalties. \$129.95

MAKE Utility (Snake) \$59.95

SUPERFONTS FOR C

Super Size Characters
Monochrome adapter
Color/graphics adapter
8 Font Libraries

Font and Function Library \$49.95

C-TERP

INTERPRETER FOR C

No Compromise, Full K&R
Built In Screen Editor
Fast, Fast Compile/Link
Use External Libraries!
Symbolic Debugging
Single Stepping
Rave Reviews!

2 disks and manual \$299.95

COMBINE AND SAVE!

SCREEN PAINTER + DATA ENTRY
BOTH for only \$179.95
(A \$260 VALUE)

C LIBRARY plus POWER WINDOWS
BOTH for only \$179.95
+ SUPERFONTS FOR C \$199.95
(A \$310 VALUE)

C BUSINESS LIBRARY
INCLUDES C FUNCTION LIBRARY, POWER WINDOWS,
SUPERFONTS FOR C, B-TREE LIBRARY, ISAM
ALL for \$299.95
(A \$440 VALUE)

C TOTAL LIBRARY
INCLUDES C FUNCTION LIBRARY, POWER WINDOWS,
SUPERFONTS FOR C, B-TREE LIBRARY, ISAM and
C-TERP C interpreter
ALL for \$499.95
(A \$740 VALUE + FREE UNIX WORKALIKE)

C DYNAMO LIBRARY #1
SCREEN PAINTER, DYNAMO DATA ENTRY, POWER
WINDOWS, C FUNCTION LIBRARY, SUPERFONTS for C
ALL for \$249.95
(A \$570 VALUE)

C DYNAMO LIBRARY #2
ALL OF C DYNAMO LIBRARY #1
PLUS B-TREE & ISAM \$349.95
(A \$700 VALUE + FREE UNIX WORKALIKE)

C DYNAMO LIBRARY #3
ALL OF C DYNAMO LIBRARY #2
PLUS C-TERP \$549.95
(A \$1000 VALUE + FREE UNIX WORKALIKE)

**FREE PC UNIX, PCVMS OR O/S TOOL BOX WITH ANY
PURCHASE OVER \$295.00 LIMIT ONE PER CUSTOMER
OFFER EXPIRES 10/31/86

SPECIAL OFFER OPERATING SYSTEMS

Multi-User Multi-Tasking Networking, Full Source

	LIST	
PCNX™ UNIX WORKALIKE	\$99	FREE*
PCVMS™	99	FREE*
O/S TOOL BOX™ (Build your own operating system)	99	FREE*

™WENDIN SOFTWARE
*with purchase of any Entelekon combination library priced at \$295 or more.
Limit 1 Special Offer item per customer.
Offer expires 10/31/86.

THE C DYNAMO FAMILY... C TOOLS THAT WORK TOGETHER

Entelekon™

SINCE 1982

12118 Kimberley, Houston, TX 77024

713-468-4412

VISA-MASTERCARD-CHECK-COD

HOW WOULD IT FEEL TO HAVE THE POWER OF TWO VAX[®] 11/780s ON YOUR DESK?

*With the Consulair[™]
Professional Development
Workstation, Now You Can!*

The programmers' dream system takes an ordinary Macintosh[™] and adds the Levco Prodigy 4, a 32 bit 68020 with a 68881 floating point coprocessor accelerated to 16 MHz, 4 megabytes of high speed memory, and SCSI interface. You get MacUser Magazine's Best Development Language of the Year, Consulair Mac C/ Mac C Toolkit with Direct Access[™] compiler support for the 68020 and 68881. Together, the system runs almost 1/2 million Whetstones per second and the Sieve benchmark in 0.68 seconds. As a special introductory offer you can upgrade your own Macintosh or Mac Plus for **\$6495**, including the complete software and hardware development system. An internal SCSI 20 Mbyte hard disk is available for an additional **\$995**.

If you want to move in smaller steps, we have a complete range of development systems to suit your needs. Leap into your future today with the Consulair Professional Development Workstation.

Call our Order and Information Hotline Today at **415-851-3272**.

Consulair

140 Campo Drive

Portola Valley, CA 94025

Circle no. 94 on reader service card.

C CHEST

Listing Two (Text begins on page 22.)

Listing 2 — b_getc.c

```

1 #include <stdio.h>
2 #include <dos.h>
3
4 /* B_GETC.C          Get a character with a direct video bios call.
5 *                  this routine can be used to complement stderr as
6 *                  it can be used to get characters from the keyboard, even when input
7 *                  redirected. The typed character is returned in the low byte of the
8 *                  returned integer, the high byte holds the auxiliary byte used to
9 *                  mark ALT keys and such. See the Technical Ref for more info.
10 *
11 *                  Copyright (C) 1985 Allen I. Holub. All rights reserved.
12 *
13 *
14 */
15
16 extern int int86(int, union REGS *, union REGS *);
17
18 /*-----*/
19
20 #define KB_INT 0x16 /* Keyboard BIOS interrupt */
21 #define GETC 0x00 /* Getc is service 0 */
22
23
24 int b_getc()
25 {
26     union REGS Regs;
27
28     Regs.h.ah = GETC;
29     int86( KB_INT, &Regs, &Regs );
30     return( (int)Regs.x.ax );
31 }
```

End Listing Two

Listing Three

Listing 3 -- look.asm

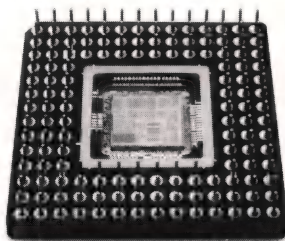
```

1 ; Static Name Aliases
2 ;
3 TITLE foo
4
5 TEXT SEGMENT BYTE PUBLIC 'CODE'
6 TEXT ENDS
7 CONST SEGMENT WORD PUBLIC 'CONST'
8 CONST ENDS
9 BSS SEGMENT WORD PUBLIC 'BSS'
10 BSS ENDS
11 DATA SEGMENT WORD PUBLIC 'DATA'
12 DATA ENDS
13 ;
14 DGROUP GROUP CONST, BSS, DATA
15 ASSUME CS: _TEXT, DS: DGROUP, SS: DGROUP, ES: DGROUP
16 ;
17 DATA SEGMENT
18 EXTRN chkstk:NEAR
19 DATA ENDS
20 ;
21 TEXT SEGMENT
22 ;
23 ; int look();
24 ;
25 ; Tests the bios to see if a key has been hit. If no key has been
26 ; hit then 0 is returned, else an int is returned in which the
27 ; high byte is the scan code and the low byte is the character
28 ; code, if the low byte is 0 then a non-ascii key has been hit
29 ;
30 PUBLIC look
31 _look PROC NEAR
32     push bp
33     mov bp, sp
34     mov ax, 2
35     call _chkstk
36
37     mov ah, 1 ; service 1, Report on character ready
38     int 016h ; BIOS keyboard interrupt.
39     jnz exit ; jump if a key is available
40             ; (return the character)
41     mov ax, 0 ; else (return 0);
42 exit:
43     mov sp, bp
44     pop bp
45     ret
46 _look ENDP
47
48 TEXT ENDS
49 END
```

End Listings

A TRAINING COURSE FOR PEOPLE WHO LUST AFTER POWER.

Intel is offering three new courses on the world's most powerful 32-bit microprocessor—the 80386. Plus a new intensive course on the 80286.



The 80386

These in-depth learning sessions are designed for engineers and programmers who want to utilize the full power and potential of these lightning-fast chips.

Lectures are combined with hands-on workshops to provide real-life situations. Allowing you to apply new concepts and techniques immediately.

Courses include:

80386 System Software

80386 Programming using ASM386

High-end Microprocessor Hardware Design

The new 80286 Microprocessor Family Course

Complete training sessions and courses can be scheduled at your facility, or at our training centers. In addition to training, Intel offers hardware/software support and consultants.

For more complete course information and schedules, call toll-free (800) 548-4725.

Or to register now, contact one of the Intel Training Centers listed below.

Intel Training Centers

Boston Area, Westford Corp. Ctr.
Three Carlisle Road, 1st Floor
Westford, MA 01886
(617) 692-1000

Chicago Area
300 N. Martingale Road, Suite 300
Schaumburg, IL 60194
(312) 310-5700

San Francisco Area
2700 San Tomas Expressway
Santa Clara, CA 95051
(408) 970-1700
Washington, D.C. Area
7833 Walker Drive, 5th Floor
Greenbelt, MD 20770
(301) 220-3380

intel[®]

The Competition is Still Multi-User System . . .

The New Tech/Mini Series from Tech Personal Computers Inc. brings the uncompromising power of XENIX SYSTEM V to your Business! It's an Intel 80286 based Xenix optimized system with multiple Intel 8088-2 I/O Coprocessors for unmatched Xenix support. The Tech 80286/Mini, with SCO Xenix Operating System, can support up to 16 Users, 16 Megabytes of RAM and is available with any

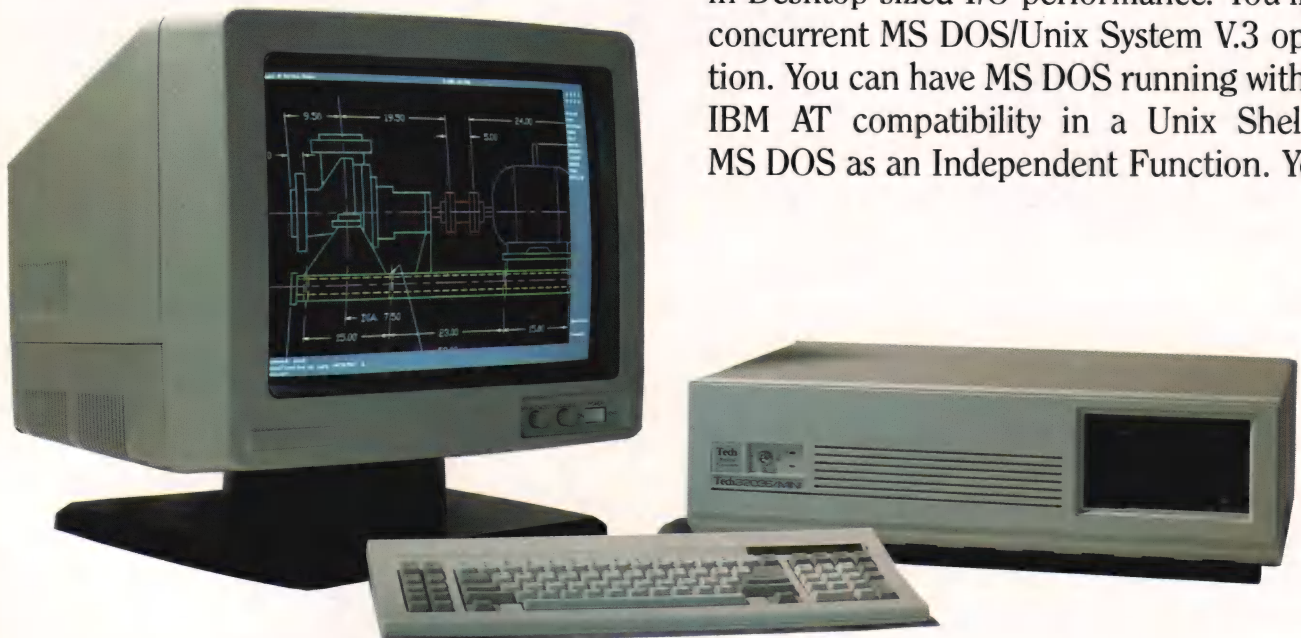
or all of the following options:

- Fixed Disks to 280 Megabytes
- Optical Mass Storage to 16 Gigabytes
- Ethernet 10 Megabyte per second Network Adaptor and Driver
- PC Net 2 Megabyte per second Network Adaptor and Driver
- 60 Megabyte Tape Back-up Cartridge Systems
- 9 Track Reel Tape Back-up Systems.

. . . They'll Never

If you want the Ultimate CAD or Multi-user Basestation, then you want our new Tech 32036/Mini. You'll get Optimized VAX 11/780

performance on your Desktop with 32 Bit 32036 NSC CPU for VAX Program compatibility and Intel 80286 CPU for the Ultimate in Desktop sized I/O performance. You'll get concurrent MS DOS/Unix System V.3 operation. You can have MS DOS running with full IBM AT compatibility in a Unix Shell or MS DOS as an Independent Function. You'll



The Ultimate CAD or Multi-User Basestation. Our New Tech 32036/Mini.

Chasing Our



*The Tech 80286/Mini shown here
with a few optional workstations.*

Tech 80286/Mini Retail Base Configurations Starting at \$2,499.00

Catch Our Mini!

have a complete Port of Unix System V.3 with Berkeley 4.2 Enhancements in Concurrent Operations. . . . and, the following features:

- 32 Bit 32036 CPU for Unix System V
- 4 MB of 32 Bit RAM
- Built-in 32081 FPU
- 16 Bit 80286 CPU for I/O and MS DOS
- 1 MB of 16 Bit RAM
- 16 MB Virtual Address Space with Demand Paged Memory
- 60 MB Tape Back-up available
- High Speed Fixed Disk to 280 MB available
- Graphics Subsystems with 1280 by 1024 line resolution and 256 colors available
- 16 Terminal support
- Ethernet 10 MB/second Network Adaptor and Driver available
- PC Net 2 MB/second Network Adaptor

and Driver available

- Optical Mass Storage to 16 Gigabytes available
- 9 Track Reel Tape Sybsystems available

TECH PERSONAL COMPUTERS is a full service manufacturer of Micro Computer Products and offers a complete line of Desktop, Portables and Multi-User Computer Systems as well as an accessory line of over one hundred enhancement products. **TECH PERSONAL COMPUTERS** are all backed by a full one year warranty with additional maintenance coverage and extended maintenance contracts available through Momemtum Service Corporation. For more information concerning hundreds of MSC Service Centers throughout the United States, contact **TECH PERSONAL COMPUTERS** at (714) 385-1711.

IBM, IBM PC, IBM XT, IBM AT are registered trademarks of International Business Machine Corporation.

(714) 385-1711

1911 Betmor Lane, Anaheim, California 92805

Telex 272006 Answer Back - TECH FAX: 7143851523

Dealers only circle **no. 245** on reader service card.

Users only circle **no. 279** on reader service card.

TECH PC

C CODE FOR THE PC

source code, of course

GraphiC 3.0 hi-res color plots	\$300
QC88 C Compiler	\$90
Concurrent C	\$45
Coder's Prolog in C	\$45
LEX	\$25
YACC & PREP	\$25
Small-C compiler for 8088	\$20
tiny-c interpreter & shell	\$20
Xlisp 1.5a & tiny-Prolog	\$20
C Tools	\$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas 78750-3409
(512) 258-0785

Free shipping on prepaid orders

No credit cards

Circle no. 250 on reader service card.

Changing Your Address?

Staple your label here.

To change your address, attach your address label from the cover of the magazine to this coupon and indicate your new address below.

Name _____

Address _____

Address _____ Apt. # _____

City _____ State _____

Zip _____

Mail to: Dr. Dobb's Journal, PO Box 27809,
San Diego, CA 92128

MEMORY RESIDENCY MADE EASY

CrackerJack Microsoftware Corporation is proud to introduce "**JACK™**", the Resident Program Developer's Kit. **JACK** contains everything you need to create your own RAM resident software without any of the headaches of memory management, windows, or DOS re-entrancy.

With **JACK** at your side, creating programs as good as, **SIDEKICK™**, couldn't be easier !!! Now you can concentrate on how your program should run, instead of worrying about how to make it memory resident.

JACK FEATURES:

- Virtually any EXISTING C or ASSEMBLER program can be made memory resident with no need for modification to the code. [PASCAL will be supported in the near future.]
- Automatic screen save on popping up and restore upon popping down.
- Your choice of Hotkey and interrupt vector for program use.
- You can use DOS function calls in your memory resident program.
- All programs developed with **JACK** will coexist with each other peacefully. No more system crashes and lost data.
- **JACK** applications can be loaded in any order and popped-up in any order. Less support problems for you and no headaches for the end user.
- Your program will beep if it is not possible for you to pop up at the time desired.
- Since **JACK** does not make any use of undocumented DOS calls, programs developed with it will not become obsolete when Microsoft releases a new version of DOS. [**JACK** requires DOS Version 2.0 or higher].

Finally, a standard for easily creating memory resident programs has arrived !

To Order, send \$199.95 [US] by certified check or international money order to:

Crackerjack Microsoftware Corporation.
200 Bay Street, PO Box 86, Toronto,
Ontario M5J-2J2 Canada.
(416) 865-9621.

Please specify your choice of the C or assembler version of **JACK** and indicate the vendor and version of your compiler. Ontario residents add 7% Ontario Sales Tax.



JACK and **CRACKERJACK** are registered trademarks of CrackerJack Microsoftware Corporation, **BORLAND** and **Sidekick** are registered trademarks of **BORLAND** International, **MS-DOS** is a registered trademark of Microsoft Corporation.

Circle no. 238 on reader service card.

NOW AT THE SBC MART COMPUTING SALE-A-THON

1.2MB Floppy on your PC or XT

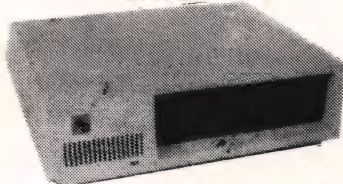
- ★ Now you can have a 1.2MB-and-360K AT-compatible floppy disk drive on your IBM (or compatible) PC or XT. Our 1.2MB controller replaces your floppy disk controller. The controller is \$149, a TEAC 1.2MB/360K disk drive is \$135, the driver is \$36. The controller combined with clock/calendar, serial and parallel ports, and able to control 3.5" and 8" floppy disk drives is only \$189.

SPECIAL BONUS Norton Utilities Ver 3.0 Reg \$99 **only \$60** w/purchase of any disk drive. IBM PC-DOS 3.2 (the real thing) Reg \$95 **only \$70**, w/purchase of motherboard.

ADD-ON CARDS

- multifunction card 384K, clk/cal w/battey, serial, parallel, and game ports, with OK save \$50 **Model MFC-4 \$89**
- monochrome graphics card runs 1-2-3 graphics, w/printer port, 720x348, **MGC-1 now \$99** (Hercules equiv.)
- floppy controller for 1-4 48tpi or 96tpi drives, w/cable save 125 **Model FDC-2 now only \$59**
- color adapter w/light pen port, RGB and composite outputs Reg \$140 **Model CC-1 now just \$79**
- ★ **better than the Super 7:** floppy disk controller, clk/calendar, serial, parallel & game ports, spooler & RAMDISK s/w **Sale \$99** ideal mate for 640K motherboards
- clock/calendar card Reg \$58, **CL-1 \$49**
- ★ **808K ON A SINGLE FLOPPY**, TEAC 96tpi 80-track disk drive kit, complete, nothing else to buy **now \$169**

CASES

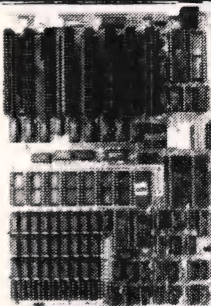


- high quality IBM PC look alike w/sideswitch, flip-flop top or side chassis
- any-combo disk drive brackets
- heavy steel 5- and 8-cutout style
- **Model CA-8** Reg \$95, **now \$69**

SBC PLEDGE

- ★ service after sale
- ★ low prices
- ★ technical support
- ★ one year warranty

MOTHERBOARDS



- full IBM PC/XT compatibility
- 8 I/O slots
- runs IBM's PC-DOS 3.1 & 3.2
- BASIC interpreter available
- great foundation for business or personal system
- 1 year warranty
- ★ **Model MB-2 640K Turbo** motherboard 4.77 & true 8MHz clock (switchable from keyboard or by software) with 256K RAM Reg \$349 **now \$239**
- **Model MB-3 640K** motherboard with OK Reg \$199 **now \$159**
- **TURBO Program** see below, right.

ABOUT OUR MOTHERBOARDS

- the most IBM PC compatible motherboards available
- each runs all commercially available software
- each works with all commercially available add-on cards

These FOUNDATION motherboards are the most compatible you can buy. All work with V20 chips, have parity checked memory, an 8087 socket, 8 adapter slots, four empty ROM sockets. Super manual includes complete data on how to put together a complete PC system. It even includes schematics.

HARD DISKS



- complete 20 MB kit
- **special only \$479**

- complete internal 1/2-high 20MB kit, incl. controller and cables, **special \$479**
- complete 30 MB kit with 1/2-high drive, uses new Adaptec 2070-A controller, **\$579**

- ★ **OMTI hard disk controller card** (this card is super fast) Reg \$220, **Model HDC-1 now \$175**, w/cables

- ★ Generalized hard-disk controller works with almost any drive. Just tell the controller your drive specs. Fast OMTI 5510-7 hard disk controller with cables and OMTIDISK utilities **HDC-4 only \$189**

- "Assembling an IBM PC/XT Compatible Computer System" A novice can put an IBM PC compatible together with this new detailed manual. **\$19**

- Microsoft's new QuickBASIC full MS BASIC compiler, Reg \$99, **only \$89**

POWER SUPPLY



- 135 Watts
- side switch
- top quality
- one year warranty
- standard cables for 4 disk drives

- **Model PS-135** power supply Reg \$129, **now only \$75**

TURBO PROGRAM

- ★ This program will set virtually all turbo motherboards to the turbo, or high-speed, mode. Can be called from the keyboard or from your AUTOTEXEC file **only \$19**

BEST DOS MANUAL

- ★ Microsoft Press' "Running MS DOS". For the less experienced and the most knowledgeable MS/PC DOS user. 423 pages. We regularly get calls from our customers telling us how great this book is. **\$21.95**

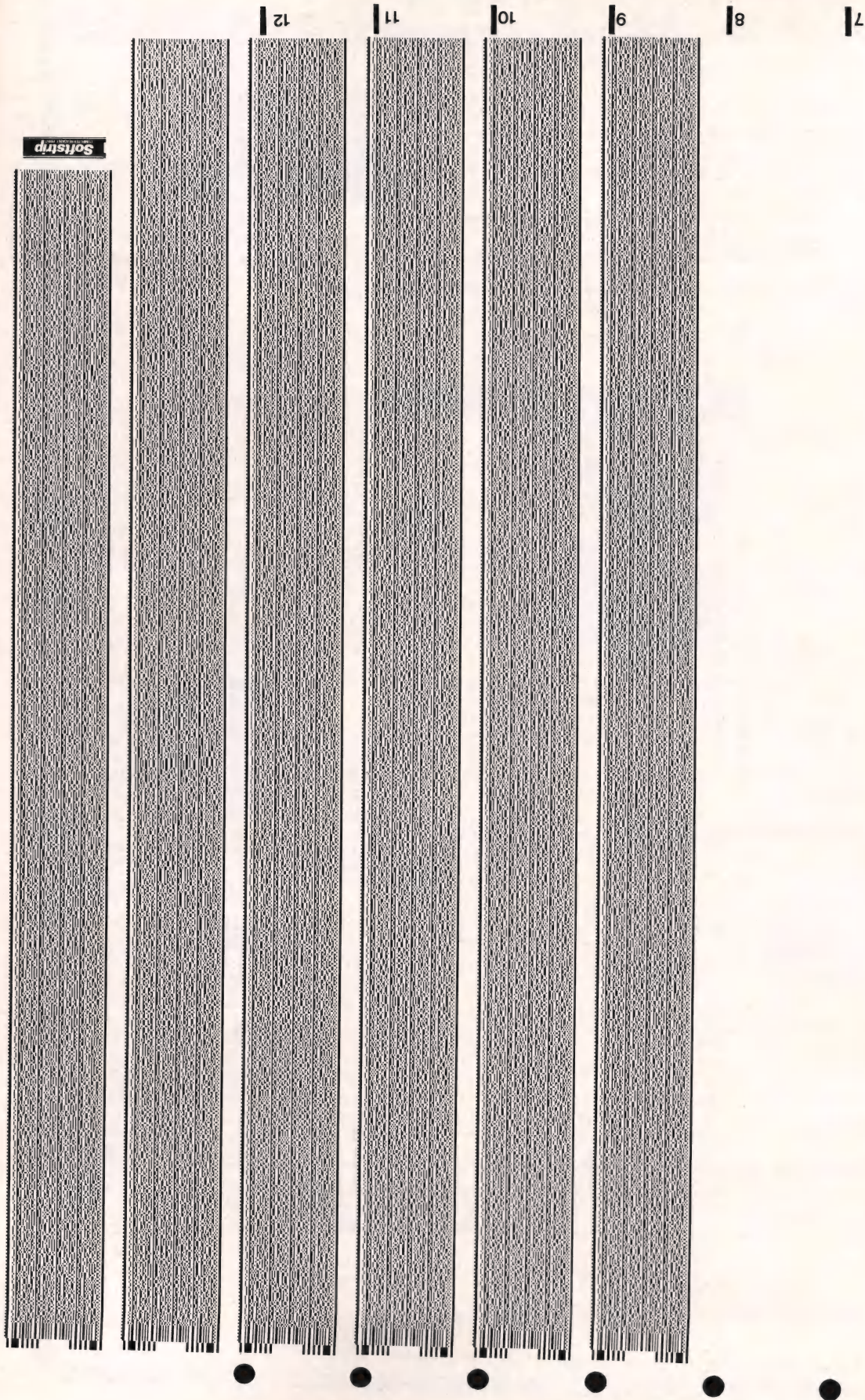
Shipping and handling: drives/motherboards/cases/power supplies **\$4.50 ea.**, cards **\$2 ea.**, software **\$2.50 ea.**, keyboards/modems **\$4 ea.**, speedup kits **\$2 ea.**, memory **\$1/set**, COMPUTERFACTS **\$3 first set then \$1/set**. CA residents add 6% sales tax.



Visa/MC/AmEx ORDERS:(619) 375-5744
The SBC MART, P.O. BOX 1296, Ridgecrest, CA 93555



The SBC Mart is part of Computing Technology, 247 Balsam Street, Ridgecrest, CA 93555



TNZ

These Softstrips by Cauzin Systems contain the listing for Richard A. Campbell's TNZ program. To read them into a computer through a Cauzin Softstrip System Reader, start on this page with strips 1-6. Then turn page 84 upside down and read in strips 7-12.

1

2

3

4

5

6

Listing One (Text begins on page 96.)

```

;
; Boyer-Moore text matching algorithm
; described in Scientific American Sept. 1984, pp. 67-68.
; Implemented for 8086 by Ray Duncan, June 1986.
;
; Call with:  DS:SI = pattern address
;            AX    = pattern length
;            ES:DI = address of string to be searched
;            DX    = length of string to be searched
;            assumes "CTAB" in same segment as pattern string
;
; Returns:   CY    = True if no match
;            or
;            CY    = False if match, and
;            ES:DI = pointer to matched string
;
boyer proc near

    mov     bp,si          ; save pattern offset
    push    di             ; save searched string offset
    push    es             ; save searched string segment
    push    dx             ; save searched string length
    push    ds
    pop     es             ; point to table with ES

    mov     cx,256          ; initialize all of table
    mov     di,offset ctab ; to length of pattern
    cld
    rep stosb

    dec     ax              ; AX = pattern length - 1
    xor     cx,cx           ; init pattern char. counter
    xor     bh,bh           ; BX will be used to index,
                           ; with char in the lower half

    bl:                                ; build table of increments
                           ; for each possible char. value
    mov     bl,[si]         ; get character

```

(continued on page 89)

FTL Modula-II

\$49.95!



Your next computer language. The successor to Pascal, Modula is powerful. Why? Once a routine is written, it need never be recompiled. Programs work everywhere from Z80 through VAX.

FTL Modula-II is a full Z80 CP/M compiler (MSDOS version soon)! It's **fast** -- 18K source compiles in 7 seconds! The built-in split screen editor is worth \$60 alone. Some standard features: full recursion, 15 digit reals, CP/M calls, coprocesses, assembler and linker. The one-pass compiler makes true Z80.COM, ROMable code, too. Get the language you've waited for now. Only \$49.95!

FTL Editor Toolkit

Full source to our split-screen programming editor. Curious? Want to customize to your tastes? Want sample Modula-II code? This is perfect for you. Comes with all you need for your personal editor or terminal installer. Just \$39.95!

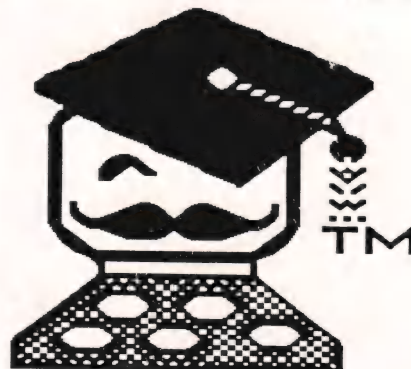
WORKMAN & ASSOCIATES
1925 East Mountain St.
Pasadena, CA 91104
(818) 791-7979

We have over 200 formats in stock! Please specify your format when ordering. Add \$2.50 per order for shipping. We welcome COD orders!

Circle no. 244 on reader service card.

MacTutor

The Macintosh Programming Journal



Simply the finest monthly technical Journal available for the Macintosh. Contains no fluff, only programs in C, Asm, Pascal, Basic & Forth. US 3rd class: \$30; US 1st class, or Canada: \$45. All Overseas: \$60. (714) 630-3730.

Subscribe
Today!

MacTutor
P.O. Box 400
Placentia, CA. 92670

Circle no. 187 on reader service card.

THE PROGRAMMER'S SHOP

31 Day
RISK-FREE TRIAL
on any product in this ad.

C Programmers: 8 Ways to Increase Productivity

Pascal to C Translation Easily and at Low Cost with The Translator (Pascal to C)

If you like Pascal's English-like syntax but want C's portability, speed, and control, or if you want to make a permanent switch to C at a very low cost, Milton Brown's Pascal to C translator is for you.

Take Jensen and Wirth standard Pascal and produce K&R standard C code. Any non-standard Pascal syntax is passed directly to the C program as in-line tokens. View the Pascal source code as the translator works. No file size limit; produces an approximately equal sized C file.

Includes meaningful, well-documented sample applications, and a manual that helps you to complete the translation. Supports Lattice, Desmet, and C86.

MSDOS \$130

FAST, Easy-to-Use Graphics, Royalty-Free: Essential Graphics

Draw fast dots, lines, circles, arcs, rectangles, and box fills. Draw a bar or exploded pie chart or a shaded line graph with one function call.

Use the font and clip-art manipulation routines with the 10 fonts included (up to 8 simultaneously), or choose from over 500 other fonts and clip-art sets available.

Essential Graphics provides fast animation and graphic windowing using GET and PUT, and generates compact code. Demonstration programs and comprehensive manual included.

Supports IBM Color, EGA, and Hercules cards, Epson and Oki printers. Lattice, Aztec, C86, Desmet, MS C, others. No royalties.

PCDOS \$219

C DYNAMO! WINDOWING: Full C Source, No Royalties POWER WINDOWS AND C FUNCTION LIBRARY

Power Windows covers all the bases: overlays, borders, 1-2-3 style or pop-up menus/help windows, zap instantly on/off screen, status lines, horizontal/vertical scrolling, color control or highlighting, word-wrap, files to windows, keyboard to windows. Powerful, easy to use, integrated error messages, thorough documentation. Supports IBM monochrome or color.

MSDOS. Only \$119.

C Function Library - includes 325 fundamental functions with readable source and thorough documentation.

MSDOS. Only \$119.

No matter what you have, you need these. Best value available. Highly recommended!

Clean, Thorough Debugging PERISCOPE I, II or IIx

Always available, start debugging after a crash. Source lines, line numbers and symbol support help save hours of frustrating work.

New version 2.1 enhancements include DOSEEDIT-like command editing, definition of up to 4 DATA windows, increased 'monitor' breakpoint speed, and much more. Other features include: disassembly, customization by YOU, in-line assembly, full 8087/287, 286, 75 + breakpoints, EGA, and traceback. It also has source support for Lattice, MSC, C86, and MS Pascal, and symbol support for Desmet and Aztec C. Even debug drivers and resident programs.

Periscope I includes an independent, memory-protected board and break-out switch; Periscope II has a break-out switch and software, Periscope IIx is software only.

PCDOS, I \$269

II \$115 through 8/31/86

IIx \$95 through 8/31/86

Add Efficient Multitasking to Your C Programs with Multi-C

Create, manage, and communicate among tasks with little RAM and processor overhead. Handle multiple users, printers, communications, or just about anything else without complex polling schemes or lockups in your programs.

Multi-C is designed for ease of use in production programming, and it **cooperates** with your operating system, so functions are **reliable**. 40+ functions use 32,000 priority levels, handle interrupts, control flags, and messages and control queues.

Multi-C is portable because it's written in C; use it for stand-alone systems, or make Multi-C the kernel of your own operating system.

Particularly good for applications in data acquisition, device control, and communications. MS C, Lattice, C86. Partial source. **No royalties.**

Cytek

MSDOS \$149

Even for Small Files: Convenient, Fast Access CBTREE

Why spend time writing file management code when you can use consistent, flexible, documented, professional functions? Even multiuser record locking and variable-length records are supported.

Add, delete, and update without needing to reindex. Store keys and record locations in B+ trees.

You can access any record or group of records by the value of a user specific key. Search your files from any point, forward or backward.

Full, balanced B-tree support includes use of multiple keys, unlimited number and length of keys.

Use this powerful ISAM, even if you've previously done without.

Learn how to write systems for managing large files by using CBTREE source as a guide. Modify it and transfer it to another operating environment without royalties.

MSDOS \$99

Flexible Screen Development with SECURITY CHECKING and HELP SCREENS: ZVIEW Screen Library

Use this field-sensitive tool to develop data entry screens and windows and provide run-time flexibility. Security level settings restrict inquiry or update of fields; multiple screen help display is available at screen and field level. You can also customize ZVIEW's operation and make any field characteristic change during execution.

ZVIEW gives you full control of attributes, colors, boxes, protected fields, scrolling, and more. Load screens from memory for fast response. Field support includes alpha, numeric, or alphanumeric data types, case conversion, range checking, and field comparison, and ZVIEW provides automatic data conversion to and from ASCII screen format. For Microsoft C, Lattice 3.0, and Aztec 3.2e. Supports EGA, color, and monochrome displays.

PCDOS \$219

Fastest C Development on Earth: Instant C version 2.0

Instant C's NEW version 2.0 gives you immediate (2 secs.) compilation and execution of large (5,000+ line) programs, and the ability to link in external (commercial or your own) libraries in an interactive, Lattice 3.0 or Microsoft 3.0 compatible, interpreter-like environment with an integrated full screen editor and source level debugger.

You'll get full K&R standard C, fast (33 second sieve) execution speed, and debugging with source code animation, single-stepping, backtracing, and unlimited conditional breakpoints.

Instant C now supports multiple screens and graphic devices, run-time checking of pointer and array references, and includes a new manual, expanded tutorial and reference section, and complete library source.

**Rational
Systems, Inc.**

MSDOS \$399

Call for a catalog, literature, advice and service you can trust

HOURS

8:30 AM - 8:00 PM EST.

800-421-8006

THE PROGRAMMER'S SHOP™
128-D Rockland Street, Hanover, MA 02339
Mass: 800-442-8070 or 617-826-7531 7/86

"We at Sunspot are thrilled to know that there is a store that can cut through all the "bull", and find us the products that most computer stores know nothing about. Keep up the good work."

Arland Hensler
Sunspot

Chalcedony PROLOG

A REAL \$99.95

Clocks in & Mellish Prolog for BOTH major microcomputing operating systems — with full cross-compatibility.

Complete with the predicates necessary for POWER AI programming:
op () name () functor () clause () =.. ("Univ")
...And no constraining data typing.

- Floating point ■ Step-by-step tutorial
- Math functions ■ Integrated editor

PROLOG

Extensible overlay library, 8087 support, large memory model (up to 640K)

PROLOG

Complete Macintosh environment with extensive pull-down menus and dialogue boxes.

No Risk Offer: Examine the PROLOG/i or PROLOG/m documentation at our risk for 30 days. If not completely satisfied, return with disk still sealed for refund.

APPLICATIONS —

Complete with SOURCE CODE

NFL X-pert \$49.95

A true interactive expert system written by a professional knowledge engineer. A valuable learning tool for any Prolog programmer interested in using Prolog to develop expert systems.

TOOLBOX \$29.95

More than 50 subroutines that speed and compress list handling, searches, sorts, and reversal algorithms. An inside look at the tricks of the professional Prolog programmer.

TOYBOX \$29.95

Written by an academician to help his students understand Prolog, this collection of puzzles and mind-teasers will illustrate how the Prolog programmer creates programs that find the best solution to the problem. Turn your computer into a super reasoning machine!

System Requirements:

Minimum 256K RAM
(320K recommended)
PC DOS/MS-DOS
ANSI Standard Support

Minimum 512K
Macintosh
Macintosh-plus and
HFS Compatible

CHALCEDONY
SOFTWARE, INC.

5580 LA JOLLA BLVD.
SUITE 126 D
LA JOLLA, CA
92037
(619) 483-8513

SAVE 10% when you buy either PROLOG/i or PROLOG/m and all 3 applications.

PHONE ORDERS: 1-800-621-0852 EXT 468

☐ PAYMENT ENCLOSED \$ _____

CA residents add 6% sales tax

☐ CHARGE MY: ☐ MasterCard ☐ Visa

Card No. _____ Exp. Date _____

Signature _____

Mr./Mrs./Ms. _____
(please print full name)

Address _____

City/State/Zip _____

PROLOG/i \$99.95
PROLOG/m 99.95
Check:
MS-DOS ☐ Mac ☐
NFL X-pert 49.95
TOOLBOX 29.95
TOYBOX 29.95
Complete Pack 188.82

SHIPPING:
\$ 5.00 U.S.
7.50 Canada
10.00 Caribbean,
Hawaii Air
20.00 Overseas Air

Not Copy
Protected



The SLR SuperLinker Plus is 3 - 10 times faster than any other linker, and look at these features:

- link a full 64K output (COM, HEX, SPR or PRL)
- works with Microsoft, Fortran, Basic, Cobol
- supports 32 character externals (SLR format)
- full drive/user support with alternate DU search
- supports 8 address spaces
- fill uninitialized spaces with 0 or FF
- global cross reference
- DSD80/SID compatible .SYM file
- manual overlays
- load map

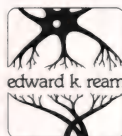
\$195

requires Z80 CP/M 2.2 or greater 32K TPA

SLR Systems

1622 N. Main St., Butler, PA 16001
(800) 833-3061 (412) 282-0864
Telex 559215 SLR SYS

Circle no. 78 on reader service card.



Transform Your Programs with CPP—C Preprocessor Plus

Includes ALL features of the standard C preprocessor.

- Define arbitrarily complex macros with #define command.
- Include and nest files to any depth with #include command.
- Include lines with #if, #ifdef and #ifndef commands.
- Define multiple constants with #enum command.
- Optional extra feature: Imbed formatting or other commands in your source code. (Lines starting with . or * are ignored.)

Fast and flexible

- 30 times faster than the Preprocessor published in Dr. Dobb's Journal.
- Can be used for any language, including assembler.
- Can be used as a stand-alone macro/include processor.
- Code can be used as the lexical analyzer for parsers or assemblers.

Complete

- You get complete SOURCE CODE in standard C.
- You get everything you need to use CPP immediately.
- CPP is unconditionally guaranteed. If for any reason you are not satisfied with CPP, your money will be refunded promptly.

Price: \$95.

Enteleki, Inc.
210 N. Bassett St., Room 101
Madison, WI 53703
Tele. (608) 258-7078

TO ORDER: Specify both the operating system (MS-DOS, CP/M 80 or CPM 68K) and the disk format (8 inch CP/M or the exact type of 5 1/4 inch disk). Send a check or money order for \$95 (\$105 for foreign orders). Foreign checks must be denominated in U.S. dollars drawn on a U.S. bank. Sorry, I do NOT accept phone, credit card or COD orders. Please do NOT send purchase orders unless a check is included.

Circle no. 150 on reader service card.

16 BIT

Listing One (Listing continued, text begins on page 96.)

```

mov     dx,ax           ; calc distance from end
sub     dx,cx           ; of pattern
mov     [bx+ctab],dl    ; put into table
inc     si              ; advance through pattern
inc     cx
cmp     cx,ax           ; done with pattern?
jne     b1              ; no, loop

pop     dx              ; restore searched string length
pop     es              ; restore searched string segment
pop     di              ; restore searched string offset
std     di              ; strings will be compared
                        ; from their ends backwards

b2:     mov     si,bp    ; get pattern addr
add     di,ax          ; point to ends of strings
add     si,ax
mov     cx,ax          ; get length to compare
inc     cx
repz    cmpsb          ; now compare strings
jz      b3              ; jump if whole string matched
inc     di              ; point to mismatched char
mov     bl,es:[di]     ; and fetch it, then
mov     bl,[bx+ctab]   ; get displacement amount
sub     di,cx           ; restore searched string address
add     di,bx          ; update searched string pointer
sub     dx,bx          ; update remaining length
cmp     dx,ax          ; enough left to compare again?
ja      b2              ; jump if searched string not exhausted
stc     ; no match, return CY=True
jmp     b4

b3:     inc     di      ; match found, return CY=False
clc     ; and ES:DI = pointer to matched string

b4:     cld                ; return to caller with direction
ret     ; flag cleared

boyer   endp

;
; Table of possible byte values: if the value exists in the pattern
; string, its byte contains its offset from the end of the pattern.
; If the value does not occur in the pattern, its byte contains the
; length of the pattern.

ctab    db        256 dup (?)

```

End Listing One

Listing Two

```

;
; General string matching routine for 8086
; (brute force version using 8086 string primitives)
; by Ray Duncan, June 1986
;
; Call with:  DS:SI = pattern address
;             AX    = pattern length
;             ES:DI = address of string to be searched
;             DX    = length of string to be searched
;
; Returns:    CY    = True if no match
;             or
;             CY    = False if match, and
;             ES:DI = pointer to matched string
;
smatch  proc    near

        mov     bp,si           ; save pattern offset
        mov     bx,ax           ; BX := pattern length
        dec     bx              ; decrement it by one

        mov     si,bp           ; AL := first char of pattern
        lodsb
        mov     cx,dx           ; remaining searched string length
        repnz  scasb            ; look for match on first char.
        jnz     s3              ; searched string exhausted, exit
        mov     dx,cx           ; save new string length
        mov     cx,bx           ; get pattern length - 1
        repz   cmpsb           ; compare remainder of strings
        jz      s2              ; everything matched
        add     di,cx           ; no match, restore string addr
        sub     di,bx           ; advanced by one char.
        cmp     dx,bx           ; searched string exhausted?
        ja      s1              ; some string left, try again
        jmp     s3              ; no match, jump to return

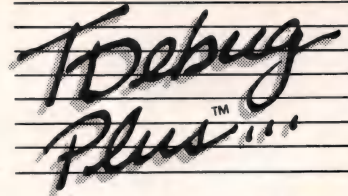
s1:     mov     si,bp           ; AL := first char of pattern
        lodsb
        mov     cx,dx           ; remaining searched string length
        repnz  scasb            ; look for match on first char.
        jnz     s3              ; searched string exhausted, exit
        mov     dx,cx           ; save new string length
        mov     cx,bx           ; get pattern length - 1
        repz   cmpsb           ; compare remainder of strings
        jz      s2              ; everything matched
        add     di,cx           ; no match, restore string addr
        sub     di,bx           ; advanced by one char.
        cmp     dx,bx           ; searched string exhausted?
        ja      s1              ; some string left, try again
        jmp     s3              ; no match, jump to return

s2:     sub     di,bx           ; match was found,
        dec     di              ; let ES:DI = addr of matched string

```

(continued on next page)

TURBO PROGRAMMERS—



... CUTS DEBUGGING FRUSTRATION.

TDebugPLUS is a new, interactive symbolic debugger that integrates with Turbo Pascal to let you:

- **Examine and change variables** at runtime using symbolic names—including records, pointers, arrays, and local variables;
- **Trace and set breakpoints** using procedure names or source statements;
- **View source code** while debugging;
- **Use Turbo Pascal editor and DOS DEBUG commands.**

TDebugPLUS also includes a special MAP file generation mode fully compatible with external debuggers such as Periscope, Atron, Symdeb, and others—even on programs written with Turbo EXTENDER.

An expanded, supported version of the acclaimed public domain program TDEBUG, the TDebugPLUS package includes one DSSD disk, complete source code, a reference card, and an 80-page printed manual. 256K of memory required. Simplify debugging! \$60 COMPLETE.

TURBO EXTENDER™

Turbo EXTENDER provides you the following powerful tools to break the 64K barrier:

- **Large Code Model** allows programs to use all 640K without overlays or chaining, while allowing you to convert existing programs with minimal effort; makes EXE files;
- **Make Facility** offers separate compilation eliminating the need for you to recompile unchanged modules;
- **Large Data Arrays** automatically manages data arrays up to 30 megabytes as well as any arrays in expanded memory (EMS);
- **Additional Turbo EXTENDER tools** include Overlay Analyst, Disk Cache, Pascal Encryptor, Shell File Generator, and File Browser.

The Turbo EXTENDER package includes two DSSD disks, complete source code, and a 150-page printed manual. Order now! \$85 COMPLETE.

TURBOPOWER UTILITIES™

"If you own Turbo Pascal, you should own TurboPower Programmers Utilities, that's all there is to it." Bruce Webster, **BYTE Magazine**

TurboPower Utilities offers nine powerful programs: Program Structure Analyzer, Execution Timer, Execution Profiler, Pretty Printer, Command Repeater, Pattern Replacer, Difference Finder, File Finder, and Super Directory.

The TurboPower Utilities package includes three DSSD disks, reference card, and manual. \$95 with source code; \$55 executable only.

ORDER DIRECT TODAY!

- **MC/VISA Call Toll Free** 7 days a week. 800-538-8157 x830 (US) 800-672-3470 x830 (CA)
- **Limited Time Offer!** Buy two or more TurboPower products and save 15%!
- **Satisfaction Guaranteed** or your money back within 30 days.

For Brochures, Dealer or other Information, PO, COD—call or write:



3109 Scotts Valley Dr., #122
Scotts Valley, CA 95066
(408) 438-8608
M-F 9AM-5PM PST

The above TurboPower products require Turbo Pascal 3.0 (standard, 8087, or BCD) and PC-DOS 2.X or 3.X, and run on the IBM PC/XT/AT and compatibles.

Circle no. 207 on reader service card.

16 BIT

Listing Two (Listing continued, text begins on page 96.)

```

clc
ret
; and return CY=False

s3:  stc
; no match,
; return CY=True

ret
smatch endp

```

Listing Three

```

( ----- )
(          C2I          )
( Convert .COM file to Inline Code )
(   by George F. Smith, 1986   )
(          )
( Sample usage:              )
( A>C2I File.Com >File.inl   )
( ----- )

($P1024,D-)

var
ctr,          ( LineSize counter )
bits : byte;  ( com file byte )
Com  : file of byte; ( com file handle )

const
LineSize = 70;
hex : array[0..15] of char = '0123456789ABCDEF';

BEGIN

Assign(Com,ParamStr(1)); ( com file name from command line )

```

```

Reset(Com);

write('Inline ( ');
ctr := 10; ( initialize counter )

While not eof(Com) do
begin
read(Com,bits); ( Get com data . . . )
Write('$', ( . . . put it inline )
hex [ bits shr 4 and $0F ], ' ');
hex [ bits and $0F ], ' ');
ctr := ctr + 5;
if ctr = LineSize then
begin
ctr := 0; ( Reset counter and )
writeln; ( start a new line )
end;
end;
Write(' '); ( Finish inline statement )
Write('^Z');
Close(Com);

END. ( C2I )

```

End Listings

IQCLISP

THE CLOSEST THING TO COMMON LISP AVAILABLE FOR YOUR PC

RICH SET OF DATA TYPES

Bignums, for high precision arithmetic
8087 support, for fast floating point
Arrays, for multidimensional data
Streams, for device-independent i/o
Packages, for partitioning large systems
Characters, strings, bit-arrays

FULL SET OF CONTROL PRIMITIVES

flet, labels, macrolet, for local functions
if, when, unless, case, cond, for conditionals
Keyword parameters, for flexibility
Multiple-valued functions, for clarity
Flavors, for object-oriented programming
Stacks, for coroutines
Closures, for encapsulation

LARGE COMPLEMENT OF FUNCTIONS

Mappers, for functional programming
format, for output control
sort, for user-specified predicates
Transcendental floating point functions
String handling functions
Over 400 functions altogether

APPLICATION SUPPORT

Save and restore full environments
User-specified initializations
Assembly language interface

HARDWARE REQUIREMENTS

8088 or 8086 CPU, MSDOS Operating System
390K RAM or more

IQCLISP

5 1/4" diskettes
and manual \$300.00
Foreign orders add \$30.00 for airmail.
U.S. Shipping included for prepaid orders.

Integral Quality

P.O. Box 31970
Seattle, Washington 98103
(206) 527-2918

Washington State residents add sales tax.
VISA and MASTERCARD accepted.

EXTENDABILITY

defstruct, to add data types
Macros, to add control constructs
Read macros, to extend the input syntax
Extendable arithmetic system
Customizable window system

DEBUGGING SUPPORT

step, for single-stepping
trace, for monitoring
break, for probing
inspect, for exploring
Flexible error recovery
Customizable pretty-printer

MSDOS INTERFACE

Random access files
Hierarchical directory access
MSDOS calls

DOCUMENTATION

On-line documentation of functions
apropos
300-page indexed reference manual

Now You Know Why **BRIEF**™ is **BEST**

"BRIEF is simple to learn and use and extremely sophisticated."

PC Magazine, July 1986

The Program Editor with the **BEST** Features

Since its introduction, BRIEF has been sweeping programmers off their feet. Why? Because BRIEF offers the features **MOST ASKED FOR** by professional programmers. In fact, BRIEF has just about every feature you've ever seen or imagined, including the ability to configure windows, keyboard assignments, and commands to **YOUR** preference. One reviewer (David Irwin, DATA BASED ADVISOR) put it most aptly, "(BRIEF)... is quite simply the best code editor I have seen."

Solution Systems™

WINDOWS

Brief does do windows, and it does them your way!

You can split the screen horizontally and vertically multiple times, creating as many windows as will fit on the screen. Each window can show any part of any file.

BRIEF'S flexible, easy to use windows make working with several files a breeze.

"BRIEF'S windows are used very effectively... You can display as many windows as you can stand at one time... Movement between windows is easy, and data can be shipped between windows. You can even edit the same program in two windows at the same time."

"You have to see this to believe it."

Elliot Niman - C Journal, Fall 1985

Every Feature You Can Imagine

Compare these features with your editor (or any other for that matter).

- FAST
- Full UNDO (N Times)
- Edit Multiple Large Files
- Compiler-specific support, like auto indent, syntax check, compile within BRIEF, and template editing
- Exit to DOS inside BRIEF
- Uses all Available Memory
- Tutorial
- Repeat Keystroke Sequences
- 15 Minute Learning Time
- Windows (Tiled and Pop-up)
- Unlimited File Size - (even 2 Meg!)
- Reconfigurable Keyboard
- Context Sensitive Help
- Search for "regular expressions"
- Mnemonic Key Assignments
- Horizontal Scrolling
- Comprehensive Error Recovery
- A Complete Compiled Programmable and Readable Macro Language
- EGA and Large Display Support
- Adjustable line length up to 512

Program Editing YOUR Way

A typical program editor requires you to adjust your style of programming to its particular requirements - NOT SO WITH BRIEF. You can easily customize BRIEF to your way of doing things, making it a natural extension of your mind. For example, you can create ANY command and assign it to ANY key - even basic function keys such as cursor-control keys or the return key.

The Experts Agree

Reviewers at BYTE, INFOWORLD, DATA BASED ADVISOR, and DR. DOBB'S JOURNAL all came to the same conclusion - **BRIEF IS BEST!**

Further, of 20 top industry experts who were given BRIEF to test, 15 were so impressed they scrapped their existing editors!

NOT COPY PROTECTED

MONEY-BACK GUARANTEE

Try BRIEF (\$195) for 30 days - If not satisfied get a full refund.

TO ORDER CALL (800-821-2492)

SOLUTION SYSTEMS, 335-D WASHINGTON ST., NORWELL, MA 02061, 617-659-1571

BRIEF is a trademark of UnderWare

THE PROGRAMMER'S SHOP

helps save time, money and cut frustrations. Compare, evaluate, and find products.

RECENT DISCOVERY

ASMLIB - 170 + Microsoft MASM compatible routines for graphics, floats, trig. Full source. PC \$135

AI-Expert System Dev't

Arity System - incorporate with C programs, rule & inheritance PC \$ 295
 Expertech - Powerful, no limit on memory size. Samples PC \$ 399
 EXSYS - Improved. Debug, file & external program access. MS \$ 339
 Insight 2+ - dB2, language. MS \$ 879
 LPA MicroProlog Intro w/APES MS \$ 149
 LPA MicroProlog Prof. w/APES MS \$ 595
 Others: Advisor (\$949),
 ES Construction (\$100), ESP (\$845),
 Expert Choice (\$449)

AI-Lisp

BYSO - Common, MacLISP compatible. 250+ functions, fast PC \$ 150
 GC LISP Interpreter - "Common", rich. Interactive tutorial Call
 Microsoft MuLisp 85 MS \$ 199
 PC Scheme LISP - by TI PC \$ 95
 TLC LISP - classes, compiler. MS \$ 225
 TransLISP - learn fast MS \$ 75
 Others: IQ LISP (\$155), UNX LISP (\$59),
 IQC LISP (\$269), WALTZLISP (\$149)

AI-Prolog

APT - Active Prolog Tutor - build applications interactively PC \$ 65
 ARITY Standard - full, 4 Meg
 Interpreter - debug, C, ASM PC \$ 350
 COMPILER/Interpreter-EXE PC \$ 795
 With Exp Sys, Screen - KIT PC \$1250
 LPA MicroProlog - intro MS \$ 99
 LPA MicroProlog Prof. -full memory MS \$ 249
 Prolog-86 - Learn Fast, Standard, tutorials, samples MS \$ 95
 Prolog-86 Plus - Develop MS \$ 250
 TURBO PROLOG by Borland PC \$ 79
 Others: Prolog-I (\$365), Prolog-2 (\$1795)

AI-Other

METHODS - SMALLTALK has objects, windows, PC \$ 69
 Q'NIAL - Combines APL with LISP. Source or binary. PC \$ 359
 Smalltalk-80 - Xerox version PC \$ 995
 Smalltalk/V-graphics PC \$ 89

FEATURES

Lattice RPG II Compiler - Run RPG II programs developed for the System III or system 32/34/36 with little or no change in source code. Screen gen, ISAM, & direct files. No royalties. PC \$639

LPA MacPROLOG - Complete incremental compiler AND an interpreter for speed, flexibility. Integrated editor, C & Mr 2 other syntax choices. Separate spaces. Optional optimizer. MAC \$295

The Programmer's Letter

to keep up with Software Trends, Technologies, and Development.

Helps you grapple with an industry that burdens you with information overload. Written from the developer's vantage point, you will get insightful information on new products, applications, and developments, and the content is complemented with a different contributing author's perspective each issue. Techniques, project management, trends in tools, pricing, industry size and interpretation are included. Request a FREE copy.

Our Services:

- Programmer's Referral List
- Compare Products
- Help find a Publisher
- Evaluation Literature FREE
- BBS - 7 PM to 7 AM 617-826-4086
- Dealers Inquire
- Newsletter
- Rush Order
- Over 700 products
- National Accounts Center

Basic

ACTIVE TRACE Debugger - BASICA, MBASIC, well liked MS \$ 79
 Basic Development System - for BASICA; Adds Renum, more. PC \$ 105
 Basic Windows by Syscom PC \$ 95
 BetterBASIC - all RAM, modules Structure. Full BASICA PC \$ 169
 8087 Math Support PC \$ 89
 Run-time Module PC \$ 235
 Better Tools - for Better Basic PC \$ 95
 CADSAM FILE SYSTEM - full ISAM in MBASIC source. MS \$ 75
 GoodBas - maintain code PC \$ 95
 LPI Basic - MS compatible UNIX \$1100
 Prof. Basic - Interactive, debug PC \$ 79
 8087 Math Support PC \$ 47
 QuickBASIC V2.0 - New interface PC \$ 95
 TRUE Basic - ANSI PC \$ 119
 Run-time Module PC \$ 459

Cobol

LPI Cobol - ANSI '74 UNIX \$1200
 Macintosh COBOL - full MAC \$ 459
 MBP - Lev. II, native MS \$ 885
 MicroFocus Professional - full PC Call
 Microsoft Cobol Tools - xref, debugger w/source support. Xenix \$359 PC \$ 239
 Microsoft Version II - upgraded. Full Lev. II, native, screens. MS \$ 479
 Realia - very fast MS \$ 839
 Ryan McFarland COBOL MS \$ 699
 COBOL-8X MS \$1049

Editors for Programming

BRIEF Programmer's Editor - undo, windows, reconfigure PC Call
 EMACS by UniPress - powerful, multifile, windows Source:\$949 \$ 299
 Epsilon - like EMACS, full C-like language for macros. PC \$ 169
 Kedit - like XEDIT PC \$ 109
 Lattice Screen Editor - multiwindow, multitasking Amiga \$100 MS \$ 109
 PC/VI - Custom Software PC \$ 129
 PMATE - power, multitask 80/86 \$ 149
 SP/PC - fast, virtual memory PC \$ 139

C Libraries-Communications

Asynch by Blaise PC \$149
 Software Horizons - pack 3 PC \$119

RECENT DISCOVERY

Test Software - capture and replay 1 complete successful test session. Automatically rerun when your code changes with Showcase. PC \$135

C Language-Compilers

AZTEC C86 - Commercial PC \$499
 C86 by CI - 8087, reliable MS \$299
 Datalight C - fast compile, good code, 4 models, Lattice compatible, Lib source. Dev's Kit PC \$ 85
 HOT C - new, intriguing PC \$ 85
 Lattice C - from Lattice MS \$299
 Mark Williams - w/debugger MS \$399
 Microsoft C 4.0- Codeview MS \$319
 Wizard C - Lattice C compatible, full sys. III, lint, fast. MS \$389

C Language-Interpreters

C-terp by Gimpel - full K & R MS \$239
 INSTANT C - Source debug, Edit to Run-3 seconds, .OBJs MS \$389
 Interactive C by IMPACC Assoc. Interpreter, editor, source, debug. PC \$225
 Introducing C - self paced tutorial PC \$109
 Run/C Professional MS \$189
 Run/C Lite - improved MS \$109

C Libraries-General

Blackstar C Function Library PC \$ 79
 C Essentials - 200 functions PC \$ 85
 C Food by Lattice-ask for source MS \$109
 C Scientific Subroutines - Peerless MS \$139
 C Tools Plus (1 & 2) PC \$149
 C Utilities by Essential - Comprehensive screen graphics, strings, source. PC \$139
 C Worthy Library - Complete, machine independent MS \$295
 Entelekon C Function Library PC \$119
 Entelekon Superfonts for C PC \$ 45
 Greenleaf Functions - portable, ASM \$139
 PforCe by Phoenix - objects PC \$299

C Libraries-Files

FILES: C Index by Trio - full B + Tree, vary length field, multi compiler /File is object only MS \$ 89
 /Plus is full source MS \$349
 CBTREE - sequential, source, no royalties MS \$ 99
 CTree by Faircom - no royalties MS \$339
 dbVISTA - full indexing, plus optional record types, pointers, Network. Object only - MS C, LAT, C86 \$159
 Source - Single user MS \$429
 Source - Multiuser MS \$849
 dBASE Tools for C PC \$ 79
 dbc Isam by Lattice MS \$199

Atari ST & Amiga

We carry full lines of Manx, Lattice, & Metacomco.
 Amiga - LINT by Gimpel Amiga \$ 79
 Cambridge LISP Amiga \$ 200
 Lattice C ST, Amiga \$ 139
 Lattice Text Utilities Amiga \$ 75
 Megamax - tight, full ST \$ 200

We support MSDOS (not just compatibles), PCDOS, Xenix-86, CPM-80, Macintosh, Atari ST, and Amiga.

THE PROGRAMMER'S SHOP

provides complete information, advice, guarantees and every product for Microcomputer Programming.

**ORDER TODAY
SPECIAL PRICE**

C-TREE by Faircom - portable, stable, thorough, balanced Btree file manager with source, MSDOS List \$395
ORDER BEFORE 10/31/86 and mention this ad for a special price of \$289
ORDER AFTER 10/31/86 for \$349

C Support-Systems

Basic-C Library by C Source PC \$139
C Sharp - well supported. Source, realtime, tasks, state system PC \$600
C ToolSet - DIFF, xref, source MS \$ 95
The HAMMER by OES Systems PC \$179
Lattice Text Utilities MS \$ 95
PC LINT - Checker. Amiga \$89 MS \$119
SECURITY LIB - add encrypt to MS C, C86 programs. Source \$250 PC \$125

C-Screens. Windows. Graphics

C Power Windows by Entelekon PC \$119
dBASE Graphics for C PC \$ 79
Curses by Lattice PC \$109
ESSENTIAL GRAPHICS - fast, fonts, no royalties PC \$219
GraphiC - new color version PC \$299
Topview Toolbasket by Lattice PC \$209
View Manager for C by Blaise PC \$219
Vitamin C - screen I/O PC \$139
Windows for C - fast PC \$159
Windows for Data - validation PC \$239
ZView - screen generator MS \$199

Debuggers

Advanced Trace-86 by Morgan Modify ASM code on fly. PC \$139
CODESMITH - visual, modify and rewrite Assembler PC \$109
C SPRITE - data structures PC \$139
DSD87 - by Soft Advances windowing, 8087 PC \$ 95
Periscope I - own 16K PC \$249
Periscope II - symbolic, "Reset Box," 2 Screen PC \$119
Pfix-86 Plus Symbolic Debugger by Phoenix - windows PC \$249
Software Source by Atron - Lattice, MSC, Pascal, Windows single step, 2 screen, log file. MS \$115
w/Breakswitch \$199

FEATURE

TurboHALO Graphics for Turbo PASCAL - respected, mature, full. 150 HALO routines, up to 16 colors, medium or high resolution, multiple fonts. IBM CGA and EGA, Hercules, AT&T DEB, more. PC \$ 99

Fortran & Supporting

ACS Time Series MS \$429
Forlib + by Alpha - graphics and file routines, comm. MS \$ 59
MACFortran by Microsoft MAC \$229
MS Fortran link to C MS \$219
No Limit - Fortran Scientific PC \$119
PolyFortran - xref, pp, screen MS \$149
Prospero - '66, reentrant MS \$349
RM/Fortran - enhanced "IBM Professional Fortran" MS \$395
Scientific Subroutines - Matrix MS \$149
Statistician by Alpha MS \$259
Strings and Things - register, shell PC \$ 59

Multilanguage Support

BTRIEVE ISAM MS \$199
BTRIEVE/N-multiuser MS \$469
CODESIFTER - Profiler. MS \$109
HALO Graphics - 115+ devices. Animation, engineering, business. Any MS language, Lattice, C86 PC \$219
PANEL - Create screen with editor, generates code. Full data validation, no royalties. Xenix \$539, MS \$229
PolyLibrarian by Polytron MS \$ 85
PVCS Version Control MS \$329
Rtrieve - Btrieve front end MS \$ 79
Screen Sculptor - slick, thorough, fast, BASIC, PASCAL. PC \$ 99
Xtrieve - organize database MS \$169
ZAP Communications - VT 100, TEK 4010 emulation, file xfer. PC \$ 95

Pascal and Supporting

ALICE - learn Pascal, Turbo compatible, interpreter PC \$ 79
MetaWINDOW - graphics toolkit PC \$119
Microsoft PASCAL - faster MS \$219
MICROTEC PASCAL - 5 memory models, "Iterators", 65 bit 8087 strings MS \$ 665
Pascal Pac with Tidy - formatter, utilities PC \$ 79
Pascal Tools - strings, screen PC \$109
Pascal Tools 2 - by Blaise MS \$ 85
Pfas - Portable Isam MS \$185
Prospero Pascal - full ISO + MS \$349
USCD Pascal - native code MS \$ 69

RECENT DISCOVERY

386 Assembler/Linker - Native or Cross Development. Full Microsoft MASM compatible plus 386/387 extensions. PC \$495

Other Languages

APL*PLUS/PC PC \$ 469
Artek ADA Compiler - DOD standard minus multitasking PC \$ 895
CLIPPER-dBASE Compiler MS \$ 449
ED/ASM - 86 by Oliver PC \$ 85
MacASM - fast MAC \$ 99
MasterForth - Forth '83 MAC or PC \$125
Microsoft MASM - faster MS \$109
Modula 2 by Volition Systems MS \$250
Modula-2/86 Compiler by Logitech w/ 8087 (\$105), 512K (\$149). PC \$ 65
Pasm - by Phoenix MS \$219
SNOBOL4+ - great for strings MS \$ 85
Turbo Edit/ASM - by Speedware PC \$ 85

Xenix-86 & Supporting

Basic - by Microsoft \$279
Cobol - by Microsoft \$795
Fortran - by Microsoft \$399
Microfocus Lev. II Compact COBOL \$899
Xenix Complete Development System \$1149

Other Products

BSW Make - like UNIX make MS \$ 85
Dan Bricklin's Demo Program PC \$ 65
dBrief - Customize BRIEF for dBASE development. with BRIEF \$275. PC \$ 95
H Test/H Format - XT Fix PC \$ 89
Interactive Easyflow-HavenTree PC \$129
Link & Locate - tools to work with Intel and Tektronix projects. MS \$329
LMK - like UNIX make PC \$149
Microsoft Windows PC \$ 75
Microsoft Windows Software Development Kit PC \$399
Opt Tech Sort - sort, merge MS \$119
PMaker - by Phoenix PC \$139
Polymake by Polytron MS \$ 79
PolyWindows Dev. Kit PC \$149
PS MAKE MS \$129
SECRET DISK by Lattice PC \$ 49
SET:SCIL - manager revisions PC \$299
Shrink/Shrinkem - put more files on disk PC \$150
SoftEst - Manage projects. MS \$350
Source Print by Aldebaran PC \$119
Synergy - Create user interfaces MS \$375
Texsys - control source MS \$ 89
Visible Computer: 8088 - Simulates demos or any .exe. com, Debugger. 350 pg. tutorial, unprotected PC \$ 69

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. All formats available. UPS surface shipping add \$3/item.

Call for a catalog, literature, advice and service you can trust



HOURS

8:30 AM - 8:00 PM EST.



800-421-8006

THE PROGRAMMER'S SHOP™
128-D Rockland Street, Hanover, MA 02339
Mass: 800-442-8070 or 617-826-7531 8/86

"YOUR SERVICE IS FABULOUS!!! . . . Your information packets are so enticing; I really want 10 of them. . . Keep up the good work."

- Dana Bredemeyer
Bredemeyer Company

STRUCTURED PROGRAMMING

Listing One (text begins on page 104.)

Listing 1: Calculating and printing percentage

```
0 CONSTANT US>D ( convert unsigned single to double )
: % ( n1 n2 - ) ( calculates and prints percentage to tenths )
  35 10 GOTOXY ( position cursor )
  10000 SWAP * / 5 + 10 / ( figure percentage and round )
  US>D <# # ASCII . HOLD #S #> ( format number as string )
  TYPE ASCII % EMIT ; ( type it with % )

( Note: Cursor-positioning is vendor dependent. )
( ASCII is immediate. It puts on the stack the )
( ASCII value of the character that follows it. )
```

End Listing One

Listing Two

Listing 2: A more general approach

```
: .0% ( n1 n2 - n3 ) ( n3 = %age n1 is of n2, rounded to tenths )
  10000 SWAP * / 5 + 10 / ;

: TENTHS ( n - adr cnt ) US>D <# # ASCII . HOLD #S #> ;

: %. ( n1 n2 - ) ?DUP ( check for 0 divisor )
  IF .0% TENTHS TYPE ASCII % EMIT
  ELSE DROP ." n/a" THEN ;
: %.R ( # n1 n2 - ) ( # is width of field; display flush right )
  ?DUP ( check for 0 divisor )
  IF .0% TENTHS ROT OVER - SPACES TYPE ASCII % EMIT
  ELSE DROP 3 - SPACES ." n/a" THEN ;
```

End Listing Two

Listing Three

Listing 3: Using CONSTANT in a defining word

```
440 CONSTANT A ( note defined by its frequency )
```

```
: OCTAVE ( creates a note of double the frequency )
  2* CREATE
  DOES> ( <adr> -- freq ) @ ;
```

```
A OCTAVE A' ( defines the frequency of the octave )
```

```
: OCTAVE 2* CONSTANT ; ( alternate definition )
```

End Listing Three

Listing Four

Listing 4: Execution array, first definition

```
CREATE OPTIONS ] >PRINTER >DISK >SCREEN >DOS [

: DO-OPTION ( n - ) 2* OPTIONS + @ EXECUTE ;

0 DO-OPTION ( to printer )
1 DO-OPTION ( to disk )
3 DO-OPTION ( to DOS )
4 DO-OPTION ( unpredictable results )
```

End Listing Four

Listing Five

Listing 5: A defining word for execution vectors

```
0 CONSTANT F -1 CONSTANT T

: VECTOR: : ( compile operators )
  DOES> SWAP 2* + @ EXECUTE ;

VECTOR: OPTION >PRINTER >DISK >SCREEN >DOS ;

0 OPTION ( to printer )
2 OPTION ( to screen )
```

End Listing Five

Listing Six

Listing 6: Bit twiddlers

```
CREATE BITS 1 C, 2 C, 4 C, 8 C, 16 C, 32 C, 64 C, 128 C,

: S>B ( ? - f ) 0<> ; ( forces to a boolean: -1 or 0 )
: MASK ( bit# - mask ) BITS + C@ ;

: AIM ( # a - bit# a ) SWAP 8 /MOD ROT + ;

: +BIT ( bit# a - ) AIM SWAP MASK OVER C@ OR SWAP C! ;
: -BIT ( bit# a - ) AIM SWAP MASK NOT OVER C@ AND SWAP C! ;
: @BIT ( bit# a - f ) AIM C@ SWAP MASK AND S>B ;
: -BIT ( bit# a - ) AIM 2DUP @BIT IF -BIT ELSE +BIT THEN ;
```

End Listing Six

Listing Seven

Listing 7: Bits for valid file name characters

```
CREATE TEST 16 ALLOT
: SETflags TEST 16 ERASE
  ASCII ! TEST +BIT
  ASCII & 1+ ASCII # DO I TEST +BIT LOOP
  ASCII ( TEST +BIT ASCII ) TEST +BIT ASCII ' TEST +BIT
  ASCII ` TEST +BIT ASCII _ TEST +BIT ASCII - TEST +BIT
  ASCII { TEST +BIT ASCII } TEST +BIT
  ASCII Z 1+ ASCII @ DO I TEST +BIT LOOP
  ASCII 9 1+ ASCII 0 DO I TEST +BIT LOOP ;

: READOUT 128 0 DO I TEST @BIT IF I EMIT THEN LOOP SPACE ;

: READ 16 0 DO TEST I + @ . 2 +LOOP ;

SETflags ok

READOUT !#$%&'()-0123456789@ABCDEFGHIJKLMNQRSTUUVWXYZ_`{} ok

READ 0 0 9210 1023 -1 -30721 1 10240 ok
```

End Listing Seven

Listing Eight

Listing 8: Checking characters

```
CREATE LEGAL 0, 0, 9210, 1023, -1, -30721, 1, 10240,

( Bit set in LEGAL only if character is legal in filename )
( Map is by ASCII value of the character. )

: OK-CHAR? ( ASCII-char -- f ; T = valid character for filename )
  LEGAL @BIT ;
```

End Listings

ATTENTION

C-PROGRAMMERS

File System Utility Libraries

All products are written entirely in K&R C. Source code included, No Royalties, Powerful & Portable.

BTree Library

75.00

- High speed random and sequential access.
- Multiple keys per data file with up to 16 million records per file.
- Duplicate keys, variable length data records.

ISAM Driver

40.00

- Greatly speeds application development.
- Combines ease of use of database manager with flexibility of programming language.
- Supports multi key files and dynamic index definition.
- Very easy to use.

Make

59.00

- Patterned after the UNIX utility.
- Works for programs written in every language.
- Full macros, File name expansion and built in rules.

Full Documentation and Example Programs Included.

ALL THREE PRODUCTS FOR —

149.00

For more information call or write:

1343 Stanbury Drive
Oakville, Ontario, Canada
L6L 2J5
(416) 825-0903

softfocus

Credit cards accepted.

Dealer inquiries invited.

the HD test

Professional Test/Format Program For Hard Drives in PC/XT/AT

- Setup interleave, step rate, etc.
- Surface analysis to flag bad tracks
- Load/save setup and bad track files
- Now supports AUTOCONFIG!
- Menu driven, with help windows
- Free PARK program included!
- Order HDTEST today for only \$99!

Proto PC inc.
612-644-4660

2424 Territorial Road, St. Paul, MN 55114
Telex 910-380-7623

Circle no. 295 on reader service card.

PC/VI

Full Screen Editor for MS-DOS (PC-DOS)

Looking for an Ultra-Powerful Full-Screen editor for your MS-DOS or PC-DOS system? Are you looking for an editor FULLY COMPATIBLE with the UNIX* VI editor. Are you looking for an editor which not only runs on IBM-PC's and compatibles, but ANY MS-DOS system? Are you looking for an editor which provides power and flexibility for both programming and text editing? If you are, then look no further because PC/VI IS HERE!

The following is only a hint of the power behind PC/VI: English-like syntax is command mode, mnemonic control sequences in visual mode; full undo capability; deletions, changes and cursor positioning on character, word, line, sentence, paragraph or global basis; editing of files larger than available memory; powerful pattern matching capability for searches and substitutions; location marking; joining multiple lines; auto-indentation; word abbreviations and MUCH, MUCH MORE!

The PC/VI editor is available for IBM-PC's and generic MS-DOS based systems for only \$149. For more information call or write:

Custom Software Systems
P.O. Box 678
Natick, MA 01760
617-653-2555

The UNIX community has been using the VI editor for years. Now you can run an implementation of the same editor under MS-DOS. Don't miss out on the power of PC/VI!

*UNIX is a trademark of AT&T Bell Laboratories.

Circle no. 268 on reader service card.

AT LAST: Professional Typesetting Capability For PC Users

With **PC_TEXTM** — the best-selling full implementation of Professor Don Knuth's revolutionary typesetting program **TEX**.

FINEST Typeset Quality Printing From:

dot matrix laser phototypesetter

$$\sum_{i=1}^{\infty} \frac{1}{i} \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \int_{-\infty}^{\infty} e^{-x^2} dx$$

WIDEST Range Of Output Device Drivers:

- Epson FX, LQ
- Toshiba
- Corona LP-300*
- Screen preview, with EGA or Hercules card
- HP LaserJet*
- Apple LaserWriter
- APS-5 phototypesetter

MOST COMPLETE Product Offering:

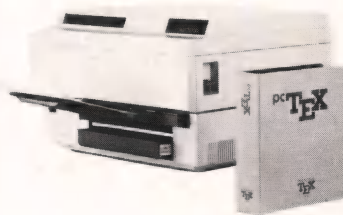
PC_TEX (not copy protected) includes the following:

- Our specially written *PC_TEX Manual*, which enables you to start using **TEX** right away.
- Custom "macro packages" that provide formats for letters, manuals, technical documents, etc.
- The **L_ATEX** document preparation system, a full-featured macro package for preparing articles, books, reports, etc., and **L_ATEX User's Manual**.
- **A_MS-TEX**, developed by the *Amer. Math. Society* for professional mathematical typesetting.

Site licenses, volume discounts, and interfaces to PC Paintbrush, PC Palette, FancyFont and Fontrix are also available.

PRICED FROM ONLY \$249.00!

(Printer drivers and interfaces additional.)



**Laser printer,
fonts & software
from \$2995.00**

For IBM PC/XT, AT or compatible, DOS 2.0 or higher, and 512K RAM. Hard disk required for printer drivers and fonts.

*HP LaserJet and Corona require additional interface boards.

For more information call or write:

Personal _TEX, Inc.

20 Sunnyside, Suite H, Mill Valley, CA 94941 (415) 388-8853

This ad, with space for the photograph, produced by PC_TEX. Typeset on the Epson FX80, the Corona LP-300 laser printer, and the Autologic APS-5 phototypesetter.

TEX is a trademark of the American Mathematical Society. Manufacturers' product names are trademarks of individual manufacturers.

Circle no. 76 on reader service card.

String Compares

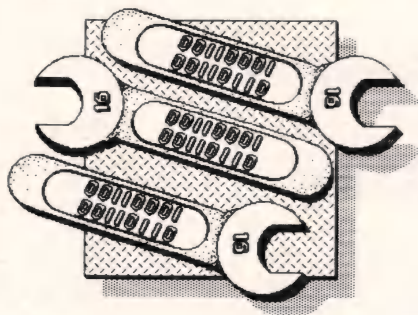
The article "Data Structures and Algorithms" written by Niklaus Wirth (who else?) in the special software issue of *Scientific American* (September 1984) included mention of an intriguing text-searching algorithm attributed to Robert Boyer and J. Strother Moore. The description of the algorithm itself is informal, but an example diagram and enough information were provided so that I could implement it in 8086 assembly language (Listing One, page 86).

The algorithm uses a clever trick to reduce the number of comparisons of the pattern against the text string being searched. At the initial entry to the searching procedure, a table is built for the pattern string in which each entry corresponds to a possible value of a member of the pattern, and the entry contains the distance from the end of the pattern of the last occurrence of that member in the pattern. If a given value does not occur in the pattern at all, its slot in the lookup table contains the length of the pattern itself. The overhead of constructing this table turns out to be insignificant when the string being searched is long.

In the main loop of the searching procedure, the comparison of the pattern with a segment of text proceeds backward from the end of the pattern. When a mismatch is found, the value in the table for the character failing the match is looked up; this value specifies the number of positions to shift the pattern forward along the text string. Thus, the mis-

by Ray Duncan

matched character is used as a pivot point, and the pattern leapfrogs its way through the text string until a complete match is found or the text string being searched is exhausted. Note that this routine could be generalized to patterns and searched strings with 16-bit elements, al-



though of course the lookup table would be quite large (128K).

Wirth seems to admire this algorithm but resorts to a little bit of hand-waving when explaining why it works. He says (page 68), "The Boyer-Moore algorithm may be faster, but can one have confidence in its correctness? In particular, how can one be certain in shifting the word [pattern] several places to the right without making any comparisons that no matching alignments were passed over? An informal argument is that a match requires identity of all the letter pairs, and the alignments passed over necessarily differ in at least one position, namely the pivot position."

Frankly, even with both working code and Wirth's explanation in hand, I am still a little perplexed with this routine. Although I had no trouble writing the code from Wirth's brief blueprint—all the pieces make sense to me, and I can trace the code and watch its operation—it still seems a little magical. A deep understanding of why it works continues to escape me.

One thing that seems evident, though, is that the Boyer-Moore algorithm is designed for processors without special string instructions. As an experiment, I coded a string search routine (Listing Two, page 89) that employs a sort of brute-force approach with the 8086's SCAS (scan string) and CMPS (compare string) instructions. It does a fast scan for a match on the first character, then performs a full string compare. In a simple test (searching the ROM space of a Compaq Portable for the string "COMPAQ"), the program in Listing Two proved to be about 30 percent faster than the pro-

gram in Listing One.

I suspect that a hybrid approach of the Boyer-Moore algorithm with the fast forward scan might give excellent results, though I will defer this exercise to a later column. By the way, the routines in Listings One and Two have purposely been made completely symmetrical in their calling conventions. If you embed these routines in other programs, you can make some further optimizations to the front end of either routine because symmetry will be of no concern to you at that point.

Resources for MS-DOS Programmers

Readers responded to my recent capsule reviews of MS-DOS programming books by suggesting the following additional references and resources:

Rollins, Dan. *IBM-PC 8088 MACRO Assembler Programming*. New York: Macmillan, 1985. \$16.95.

Jourdain, Robert. *Programmer's Problem Solver for the IBM PC, XT, and AT*. New York: Brady Publishing (Simon and Schuster), 1986. \$22.95

Generic PC-DOS newsletter about PC-DOS on non-IBM systems, published by Fred Greeb, 8403 W. Illiff Lane, Lakewood, CO 80227.

Assembly Language Supplement Newsletter, published by William J. Claff, 7 Roberts Road, Wellesley, MA 02181.

I will include more detailed descriptions of these books and newsletters in future columns, after I have seen them myself.

In-Line Assembly for Turbo Pascal

George F. Smith of Lilburn, Georgia, writes: "Some of your readers may be writing assembler routines for Turbo Pascal, as I have done in my user-supported package Boosters. Routines written in assembler for Turbo Pascal may run as external

.COM files or in-line code. Once a routine is running properly, I like to use it as in-line code because it compiles at maximum velocity and doesn't bother the disk drives.

"I'm enclosing a utility program, C2I [Listing Three, page 90], that makes it easy to get from .COM to in-line code. The program reads a character from a file, converts each nibble to hex/ASCII, then applies formatting for syntax requirements and token readability. It writes the result to standard output and repeats this process until it reaches an end of file.

"To run C2I, convert it to a .COM file first, then from DOS type:

```
A>C2I filename.COM > filename.INL
```

Filename.COM, of course, must be a machine-code file that works as a Turbo Pascal function or procedure and that you understand how to use. When C2I finishes, filename.INL will contain the in-line code.

"To merge the generated in-line code file into your Pascal program, read filename.INL into the Turbo editor using Ctrl-K-R, then add header and trailer information (here assuming a procedure):

```
Procedure Some ( parameters . . . );
begin
    InLine (
        $1E . . .
        . . .
        /$1F );
end;
```

"A little doctoring is usually necessary before the in-line routine will work properly. The .COM files Turbo Pascal uses as externals usually begin with the sequence:

```
Push    BP        ;/$55
Mov     BP,SP     ;/$8B/$EC
```

and end with

```
Mov     SP,BP     ;/$8B/$E5
Pop     BP        ;/$5D
Ret     ;/$C2/$00/$00
```

The corresponding object code as it appears in the in-line file is shown above on the right. Turbo Pascal provides this code for you when it compiles the routine's header and termi-

Is LISP Right for Your Expert System?

Find Out — FREE

You can explore LISP by examining a complete sample problem. Call and we will send you a free source listing of "SELECTWP".* It prompts users for criteria and helps them choose which micro word processor to buy.

Look over the TransLISP syntax (COMMON LISP compatible). Your application will probably have similar characteristics.

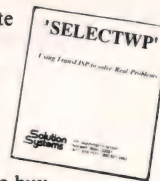
Power & Flexibility

Do you get flexibility in PASCAL and C? Of course, but examine the listing of SELECTWP to see how much more power and flexibility you get. The LISP advantages:

- forward references make program flow fit the problem
- manipulate data structures of varying sizes
- create your own language to fit the problem domain
- avoid mundane, busy work required with traditional procedural languages
- powerful function and macro building facilities provide better data abstraction

Full refund if not satisfied in 30 days.

*SELECTWP includes 512 lines of LISP code and 335 lines of comments.



TransLISP gives You the Advantage

Using TransLISP for your expert system has several advantages over other AI tools. And you will see SELECTWP illustrate:

- * the ability to control how decisions are made
- * the freedom to assign weights and react to user choices
- * the complete control you have over how a problem is solved, and interaction with the user

Nothing to lose

Examine LISP carefully by studying a practical program free.

Or buy TransLISP risk free. SELECTWP is just 1 of over 20 sample programs in the complete TransLISP system. The other sample programs include: an adventure game, a program to read dBASE SDF files, "Job Counselor" and more. Use the modular tutorial, the complete 300+ function LISP interpreter, and the online help, to get started in LISP in only a few hours.

Develop programs of up to 12000 lines on a 640K system or use TransLISP on a floppy only, 256K RAM machine. MSDOS.

Call 800-821-2492 for SELECTWP FREE. Or order the complete TransLISP system risk free for only \$95.

Solution Systems

335 Washington St., Norwell, MA 02061 (617) 659-1571

Circle no. 148 on reader service card.

The C Programmer's Assistant

C TOOLSET

UNIX-like Utilities for Managing C Source Code

No C Programmer should be without their assistant - C ToolSet from Solution Systems. The package consists of several utilities designed to help make C programming tasks easier.

C ToolSet (formerly C Helper) includes:

DIFF - Compares text files on a line-by-line basis or use CMP for byte-by-byte - indispensable for showing changes among versions of a program under development. So "intelligent" it stays in synch even when you add 100 lines.

GREP - Regular expression searches - ideal for finding a procedural call or a variable definition amid a large number of header and source files.

FCHART - Traces the flow of control between the large modules of a program.

PP (C Beautifier) - Formats C program files so they are easier to read.

XREF (CCREF) - Cross references variables from a program.

Available For MS-DOS - \$95

ONLY \$95

Source Code Included

Solution Systems TM

335 Washington St.
Norwell, MA 02062
617-659-1571

800-821-2492

Full refund if not satisfied during first 30 days.

Circle no. 152 on reader service card.

Order our FULL C COMPILER For \$59⁹⁵ and we'll give you a free CED Program Editor



The Ecosoft Eco-C88 compiler for the 8088 and MSDOS is going to set a new standard for price and performance. Consider the evidence:

Compiler	Eco-C88	Lattice (1)	C86 (1)
Seive	12	11	13
Fib	43	58	46
Deref	14	13	—
Matrix	22	29	27
Price	\$59.95	\$500.00	\$395.00

(1) *Computer Language*, Feb., 1985, pp. 73-102. Reprinted by permission.

Eco-C88 Rel. 3.0 on IBM PC with 2 floppy disks, 256K. Benchmarks from Feb., 1985, *Computer Language*.

Eco-C88 includes:

- ★ All operators and data types (except bit fields)
- ★ Prototyping, structure passing and assignment, enum and void language enhancements.
- ★ Tiered error messages (gives you selectable levels of "lint" semantic checking)
- ★ memfiles (TM) for using memory outside the 128K limit as a file
- ★ Expanded library with over 200 functions (many of which are System V compatible) plus color and transcendentals
- ★ ASM or OBJ output; uses the MSDOS linker
- ★ 8087 support with 8087 sensed at runtime
- ★ cc and "mini-make" for easy compiles (with source code)
- ★ expanded user's manual

If ordered with the compiler, the C library source code (excluding transcendentals) is \$25.00 and the ISAM file handler (as published in the *C Programmer's Library*, Que Corp.) in OBJ format is an additional \$15.00. Please add \$4.00 for shipping and handling. To order, call or write:



Ecosoft, Inc.
6413 N. College Avenue
Indianapolis, IN 46220

(317) 255-6476 • 8:30-4:30
1-800-952-0472
(orders only)



Circle no. 89 on reader service card.

MS-DOS, UNIX, APPLE MAC, CP/M, NETWORKS and MORE. ONE c-tree ISAM DOES THEM ALL!

The creator of Access Manager™ brings you the
most powerful C source code, B+ Tree file handler: **c-tree™**

- multi-key ISAM and low-level B+ Tree routines
- complete C source code written to K&R standards
- single-user, network and multi-tasking capabilities
- fixed and variable record length data files
- virtually opened files accommodate limited file descriptors
- no royalties on application programs

\$395 COMPLETE

Specify diskette format:

- 5¼" MS-DOS
- 8" CP/M
- 3½" Mac
- 8" RT-II



For VISA, MC and COD orders
call (314) 445-6833

FairCom

2606 Johnson Drive
Columbia, MO 65203

© 1985 FairCom

The following are trademarks: c-tree and the circular disk logo—FairCom; MS—Microsoft Inc.; CP/M and Access Manager—Digital Research Inc.; Unix—AT&T; Apple—Apple Computer Co.

Circle no. 93 on reader service card.

16-BIT

(continued from page 97)

nating *END* statement. If you edit out these bytes from the in-line code, it should behave as well as the external .COM file."

WINDOW.ASM Revisited

Chris Dunford, one of the sypops of the IBM PC SIG on CompuServe, has some useful comments and suggestions regarding John Seal's WINDOW.ASM program that was published in the May 1986 16-Bit Toolbox column.

"First, the EGA adds BIOS video services 10h, 11h, and 12h, so the program won't run on an EGA-equipped PC. Better stated, it would probably run, but my guess is that it would fail the 'already installed' test and refuse to install itself. If it did install, then those EGA functions wouldn't be available to other programs. I realize that WINDOW.ASM was probably written before the EGA was available.

"Second, statements such as 'All registers preserved except ax' (in the prologue to the *set_window* routine) may be misleading. WINDOW.ASM uses some standard BIOS video services, which do not guarantee to preserve *si*, *di*, and *bp*. Those registers are usually returned unchanged on a standard PC (but not always—check *bp* after a BIOS scroll); however, some compatibles do use them more extensively. I know of one programmer who came to grief by actually testing to see whether *si* was affected by a particular operation on his PC. It wasn't, so he saved himself the 2 bytes of a push/pop and everything ran fine—until the program was executed on one particular compatible. *Si* was altered, the program failed, and it took him forever to figure out why.

"Finally, in checking for command-line parameters, WINDOW.ASM scans the full 127-byte unformatted parameter area beginning at PSP:0081h looking for a (. This is not safe because there is no guarantee that the area following the actual parameters has been zeroed by the MS-DOS loader: It could easily contain junk left behind after the execution of some other program. The length of the actual command tail passed to the program is available at PSP:0080h and should be used when scanning for ar-

MODULA-2

In no other language do off-the-shelf software components work so well. **REPertoire**, the accepted choice of *Modula-2* developers, is a proven, reliable set of components designed exclusively for *Modula-2*.

\$19

A shockingly aggressive price.

Includes fully usable SYM and LNK files, but no source. Comes with manual on disk; add \$15 for each printed manual. Upgrade to full source version for \$79.

With printed manual and full source code (440K),..... ONLY **\$89**

INCLUDES:

- ★ A sophisticated screen design/display system: Instant display of full-color screens that obtain and check input, provide help, scroll within windows, and intelligently adapt to the hardware.
- ★ A natural-language analysis system integrated with the screen system's input-checking functions.
- ★ Over 200 low-level routines, including a generic list-handling module.
- ★ NEW with Release 1.3: DBMS with true variable-length keyed records.
- ★ A multi-process, multi-window editor module.
- ★ A thoroughly indexed, 250-page manual.

For IBM compatibles. Versions for Logitech, ITC, and other compilers. No royalties. Call for free demo and complete manual (on disk).

PMI
4536 S.E. 50th
Portland, OR 97206

The leading supplier of *Modula-2* software components.

VISA/MC/COD
(503) 777-8844 (24 hrs.)
In Europe, call 47-06-97-00-40
Compuserve: 74706, 262
BIX: pmi

Circle no. 239 on reader service card.

ASMLIB

Assembly Language Programming Library
for the IBM PC/XT/AT or compatible DOS systems.

ASMLIB gives the Assembly Language programmer 190+ assembly functions which do -

- Graphics functions draw CIRCLES, ARCS, ELLIPSES, LINES, and plots POINTS on the Color Adapter, Enhanced Graphic Adapter, and the Herc. Monochrome Card. Functions also allow PAINTING, IMAGE SAVES and RESTORES, and SCROLLING.
- Text Windows - Up to 64 text windows may be defined, outlined, overlapped, moved, and can be grouped onto 256 logical display pages.
- Floating Point - Arithmetic and Trigonometry functions for the MS and IEEE (8087) floating point formats (both 4 and 8 byte precision), and the 8087 (80287) can be utilized automatically if installed into the target system.
- ASCII String/Numeric conversion routines provide a user interface to the math functions. ASFORMAT function allows numeric values to be formatted utilizing commas, dollar signs, left or right justified, etc (i.e. BASIC's PRINT USING).
- Mouse Support - ASMLIB provides support for any mouse device which adheres to the MS Mouse software standard.
- Dynamic Memory support can utilize all available memory (up to 640k). Blocks of memory can be allocated, changed, moved, and killed.
- Console I/O, Disk I/O with file copy routines, Asynchronous Communications, Printer Support, ASCII String support, Sound generation, plus much more.

ASMLIB is supplied with complete MASM source code on 3 DOS diskettes along with a 215+ page reference manual.

**- ALL FOR ONLY -
\$149.00**

For ordering or info please call:

BC Associates
13073 Springdale St., Suite 134
Westminster, CA 92683
(714) 741-3015

Phone COD orders accepted - ORDER YOURS TODAY!!!

Circle no. 182 on reader service card.

C Cross Compiler 68000/08/10/20

Features:

- Full, Standard C
- Easy to Use Compiler Options
- Complete User Documentation
- Global Code Optimization
- Optional Register Allocation Via Coloring
- ROMable and Reentrant Code
- Comprehensive Royalty Free Run-time Library
- Floating Point Library Routines
- Intermix MCC68K C with ASM68K Assembly Language or Microtec PAS68K Pascal
- Optional Assembly Language Listing Intermixed with MCC68K C Source Line Number
- Symbolic Debug Capability

The Microtec MCC68K C Cross Compiler is a complete implementation of the 'C' programming language as defined in The C Programming Language by Kernighan and Ritchie with extensions.

MCC68K emits highly optimized assembly language code for the Microtec ASM68K Motorola compatible assembler.

The Microtec MCC68K package includes the compiler, relocatable macro assembler, linking loader, run-time library, and comprehensive user's guide.

Now Generates:
Position Independent Code

Host computers include: DEC VAX, DG MV-Series, Apollo, IBM PC and PC-compatibles..

We're **Functional** and **Fast** and **Serious** about our products. We've been providing flexible and economical solutions for software developers since 1974.

Beginning with product concept, through development, quality assurance, and post-sales support - **Quality, Compatibility and Service** are the differences which set Microtec Research apart from others.

If you're a serious software developer, shopping for software development tools, call or write today for more information:

800-551-5554,
In CA call (408) 733-2919.

3930 Freedom Circle, Suite 101, Santa Clara, CA 95054
Mailing Address: P.O. Box 60337, Sunnyvale, CA 94088

**MICROTEC®
RESEARCH**

At last! - Fast, On-screen FLOWCHARTS

Finally! An on-screen flowchart processor that knows about flowcharts - not just another "screen draw" program that makes you do most of the work.

Interactive EasyFlow is a powerful full-screen graphics program dedicated to flowcharts and organization charts. With this program you can quickly compose charts on the screen. More important, you can easily modify charts so they are always up to date.

Features: • Text is automatically centered, character by character, within shapes as you type it • Text formatting controls allow you to over-ride the automatic formatting where desired • Lines are created by specifying the starting and ending points - the program automatically generates the route • Cut and paste facility allows arbitrary chart fragments to be moved, copied rotated, reflected or sent to/from disk • Shape insert-delete and row/column insert-delete • Charts can be up to 417 characters wide by 225 lines high. Charts too wide for the printer are automatically printed in strips. • Charts can be larger than the screen - the window into the chart scrolls both horizontally and vertically as necessary • Works with many popular matrix printers including Epson, IBM graphics printer and compatibles. Full support for HP LaserJet and LaserJet Plus. Works with

HP 7475A (& compatible) plotters. Can be used with ANY printer when non-graphic (character) output is acceptable

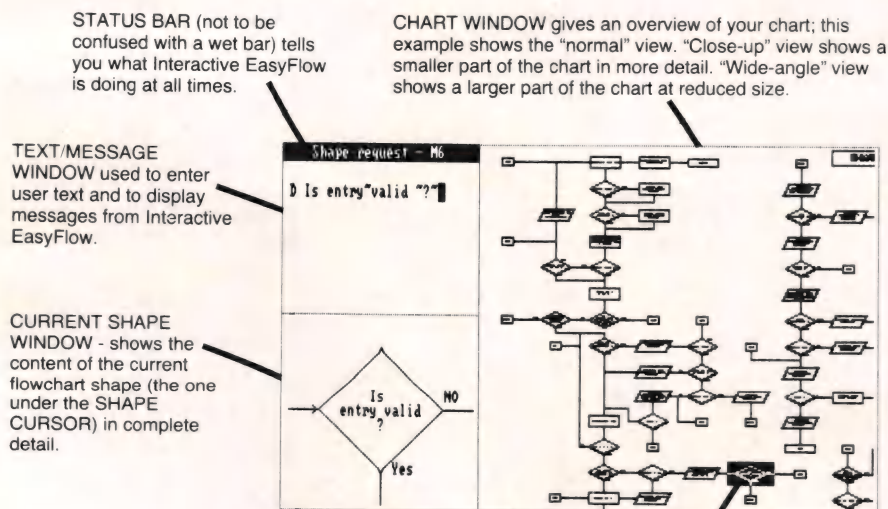
• All standard flowcharting shapes included • Most shapes supplied in large, medium and small sizes • Extensive manual (100+ pages) includes many examples • Context sensitive "help" facility provides immediate assistance at any time • Any number of titles can be placed on a chart • Commentary text blocks can be placed anywhere in the chart • Fast: written in assembly language • Plus many more features than we can mention here

Requires at least 320K memory, DOS-2 or higher and an IBM or Hercules compatible graphics card. On EGA, full 640x350 resolution is used.

Order direct for only **\$149.95** + \$2.00 S&H (USA/Canada), \$10.00 (foreign). Payment by MO, check, VISA, MasterCard, COD or Company PO. Rush orders accepted (\$15.00 S&H; USA/Canada only). Rush orders received by noon will be delivered the next business day (to most locations).

Order Desk: **1-800-267-0668**

The sample screen display shown below is typical of what you see while editing a chart. Other screen displays are provided for entering titles, changing options, getting "help" and so on.



SHAPE CURSOR shows where you are in the chart. Cursor keys move it around; chart window scrolls if you run off the edge of the window.

Circle no. 176 on reader service card.

HavenTree Software Limited
P.O. Box 1093-B
Thousand Island Park, NY 13692
Information: (613)544-6035 ext 26

16-BIT

(continued from page 98)

guments passed to a program."

Modifying the Master Environment Block

Whenever MS-DOS programmers get together to talk about the subjects that really aggravate them, the machinations that are necessary to modify the system's master environment block are always high on the list of topics.

The environment block is a paragraph-aligned data block that contains a series of ASCII (null-terminated) strings, the whole set of strings being terminated by an additional null byte. Each string is in the form:

variable=parameter

Under DOS Versions 2 and 3, three particular variables—*COMSPEC=parameter*, *PATH=parameter*, and *PROMPT=parameter*—are always found in the environment block. These are initialized during the system boot process and tell COMMAND.COM where to find the transient portion of itself for reloading, the subdirectories to search for executable files, and the format of the user prompt, respectively. These three environmental variables may be modified, and new variables may be added, by SET commands entered at the DOS command level.

The environment block can be as large as 32K and can be a very effective means of passing "global" configuration information to executing programs. The Microsoft C compiler and Microsoft linker, for instance, use environmental variables to find include and object library files. You would also think that, because the environment block can be so large, it would also be a very nice way to pass data between sequentially executing programs—a sort of built-in system "COMMON." Although simple in concept, this kind of use of the environment block turns out to be very difficult in practice.

A pointer to the environment block for a given process is found at offset 002ch in that process's program segment prefix under current versions of DOS. This is not a pointer to one, centralized environment block for the system, however, but is a pointer to a

FORTRAN PROGRAMMERS

Looking for the right PC FORTRAN LANGUAGE SYSTEM? If you're serious about your FORTRAN programming then you should be using F77L- LAHEY FORTRAN.

Editor's Choice - PC Magazine

- Full FORTRAN 77 Standard (F77L is not a subset)
- Popular Extensions for easy porting of minicomputer and mainframe applications
- COMPLEX*16, LOGICAL*1 and INTEGER*2
- Recursion - allocates local variables on the stack
- IEEE - Standard Floating Point Arithmetic
- IMPLICIT NONE
- Long variable names - 31 characters
- Fast Compile - Increase your productivity
- Source on Line Debugger (Advanced features without recompiling)
- Arrays and Common Blocks greater than 64K
- Clear and Precise English Diagnostics
- Compatibility with Popular 3rd Party Software (i.e. Lattice C)
- Easy to use manual
- Technical Support from LCS

• NEW FEATURE - NAMELIST

F77L - THE PROGRAMMER'S FORTRAN \$477.00 U.S.

System Requirements: MS-DOS or PC-DOS, 256K, math coprocessor (8087/80287)

FOR MORE INFORMATION: (702) 831-2500



Lahey Computer Systems Inc.

P.O. Box 6091 Incline Village, NV 89450/USA.

International Dealers:

England:	Grey Matter Ltd.,	Tel: (0364) 53499
Denmark:	Ravenholm Computing,	Tel: (02) 887249
Australia:	Computer Transitions	Tel: (03) 537-2786
Japan:	Microsoft, Inc.,	Tel: (03) 813-8222

SERVING THE FORTRAN COMMUNITY SINCE 1967

Circle no. 186 on reader service card.

TOTAL CONTROL with LMI FORTH™



For Programming Professionals:
an expanding family of compatible, high-performance, Forth-83 Standard compilers for microcomputers

For Development:

Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler

- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

Call or write for detailed product information and prices. Consulting and Educational Services available by special arrangement.

LMI Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to: (213) 306-7412

Overseas Distributors.

Germany: Forth-Systeme Angelika Flesch, Titisee-Neustadt, 7651-1665

UK: System Science Ltd., London, 01-248 0962

France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16

Japan: Southern Pacific Ltd., Yokohama, 045-314-9514

Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

C for yourself!

A full year for only \$18.

Think about it, a full year of technical and useful information about C. **The C Journal** provides programming information for any machine - IBM PC™, UNIX™ -based, Macintosh™, or CP/M™ - micro, mini or mainframe. Look forward to each issue for:

- NEW in-depth reviews and feature articles - C compilers, editors, interpreters, function libraries and books.
- NEW efficiency hints and tips.
- NEW interviews with C experts.
- NEW news and rumors from the ANSI standards committee and industry.

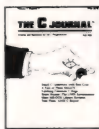
Subscribe today to the only magazine that is dedicated specifically to C - **The C Journal**.

Please send check or money order for \$18 (cover price \$28) to:

InfoPro Systems

3108 Route 10, Denville, NJ 07834
Call TOLL FREE (800) 628-2828 ext. 849
(for charge card orders only)

Please add \$9 for overseas mail and \$6 for Canadian subscriptions.



THE C JOURNAL™

Trademarks - IBM PC: IBM Corp.; UNIX: AT&T Bell Labs; Macintosh: Apple Computer Corp.; CP/M: Digital Research Inc.; **The C Journal**: InfoPro Systems.

Circle no. 194 on reader service card.

Circle no. 205 on reader service card.

static copy of the environment block of the parent program that caused the current process to be executed. The parent program may be the system's command processor (usually COMMAND.COM), but it may also be any other process that can perform an EXEC call (int 21h, function 4bh). Changes made by a program to its own environment block are visible only to other programs that it spawns

explicitly and have no effect on its own parents or on programs that execute after it terminates.

The environment block for a given process sits inside a memory block (memory arena) that has been allocated by the system loader via the MS-DOS allocate memory block function (int 21h, function 48h), and the program code and data for a process sit inside another such block. Each allocated memory arena is controlled by a 16-byte memory control block (called an arena header), which sits

immediately below it. The control blocks contain three items of useful information: a byte designating whether the control block is a member or the last in the chain of all control blocks, the segment of the PSP of the program "owning" the allocated memory block (this slot is zero if the block is free), and the length of the allocated block in paragraphs. Thus, the control blocks are chained implicitly because you can jump from one control block to the next toward high memory with the length information that each contains.

Because the chain of memory control blocks can be followed in only one direction, and your program is usually sitting at or near the end of the chain, there is no well-behaved (for that word read *documented*) way to trace back through the allocated memory blocks toward low memory and find the master environment block owned by COMMAND.COM or another shell. At first glance then, it seems impossible for your program to affect the master environment in a way that will pass information to all other programs that are executed. Hackers will be hackers, though, and there are at least three ways to accomplish the effect of a SET command against the master environment block from the level of an executing application. I'll discuss these techniques in next month's column. See you then!

DDJ

(Listings begin on page 86.)

Vote for your favorite feature/article.
Circle Reader Service No. 6.

WIZARD C

"The Best Compiler Today"

"Wizard's is the best compiler today. What it does have is library source for a very large library, good documentation, excellent support, and lint.

"Our choice if we could make our own? We would take Wizard . . ."

Dr. Dobb's Journal
August, 1986

"... the compiler's performance makes it very useful serious software development."

PC Tech Journal
January, 1986

"... written by someone who has been in the business a while. This especially shows in the documentation."

Computer Language
February, 1985

You've Tried The Rest
Now Try The Best

(617) 641-2379

Only \$450



WIZARD
SYSTEMS SOFTWARE, INC.

11 Willow Court, Arlington, MA 02174

Circle no. 116 on reader service card.

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

CAF 8 MHZ AT COMPATIBLE - \$1399

- * PRODUCED BY CAF: A \$500 MILLION/YEAR TAIWANESE CONGLOMERATE.
- * BURNED IN FOR 48 HRS. * INCLUDES DOS 3.1 AND MANUALS.
- * 512K RAM, EXPANDABLE TO 1MEG ON M.B. * 1.2 MEG FLOPPY INCLUDED.
- * 80286 WITH OPTIONAL 80287 * ON BOARD REAL TIME CLOCK.
- * AT STYLE KEYBOARD WITH ENLARGED ENTER AND SHIFT.
- * 195W POWER SUPPLY * COMPLETE SERVICE MANUAL AVAILABLE: \$40.
- * COMPLETE SYSTEM DOCUMENTATION INCLUDED, PLUS DOS MANUALS.
- * FLOPPY AND HARD DISK CONTROLLER CARD INCLUDED.
- * ERSO BIOS FOR MAXIMUM AT SOFTWARE COMPATIBILITY.
- * LIMITED 1-YEAR FACTORY WARRANTY. * FCC CERTIFIED!
- * 1.2 MEG FLOPPY CAN READ AND WRITE 360K DISKETTES. ADD \$15 UPS.

CMI 6640 33 MB. HARD DISK 39 MS - AT COMPATIBLE. VERY LIMITED STOCK — \$325

MEX-PC MODEM SOFTWARE PACKAGE \$69.95

A FANTASTIC COMMUNICATIONS PACKAGE WITH FEATURES TOO NUMEROUS TO LIST. SEE MAY/JUNE '86 MICRO SYSTEMS JOURNAL FOR FULL REVIEW OR CALL FOR BROCHURE. SUPPORTS COLOR, KERMIT PROTOCOL, Y MODEM BATCH, VT100 EMULATION, REMOTE OPERATION, ON LINE HELP, ETC. RUNS UNDER PC/MS DOS. 180 PAGE MANUAL & BINDER.

MEX-PACK LIST IS \$99.95 — SPECIAL \$69.95

STANFORD TURBO XT - \$399

- 4.77 OR 8 MHZ * FCC CERTIFIED * 640K MOTHER BOARD (O K) * .135 WATT P.S.
- * XT-AT 5160 KEYBOARD * FLIP TOP CASE * 8 EXPANSION SLOTS * ASSEMBLED AND TESTED * WITH BIOS * FLOPPY CONTROLLER CARD & CABLES * WITH DOCUMENTATION. ADD \$12 UPS.

A.R.C. XT TURBO - \$529

- 4.77 OR 7.4 MHZ * 8 EXPANSION SLOTS * SLIDE OUT PC CASE * 5160 XT-AT KEYBOARD * MS DOS 3.1 * BASIC DISKETTE * 256K RAM, EXPANDABLE TO 640K
- * 4 LAYER JAPANESE MOTHER BOARD * FULLY TESTED AND BURNED IN.
- * SUPERIOR, CLEAN BIOS. * 135 WATT P.S. * SUPERB DOCUMENTATION * FCC CERTIFIED. ADD \$15 UPS.

XT OPTIONS:

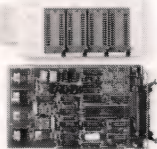
- * V20.8 MHZ NEC PROCESSOR. REPLACES 8088. - \$12.95
- * TWO 1/2 HT. D.S. DD. FLOPPIES INSTALLED: \$175/Pair
- * 256K DRAM. 150 NS. INSTALLED AND TESTED: \$28/9
- * 64K DRAM. 150 NS. INSTALLED AND TESTED: \$10/9
- * HARD DISK: ST225 WITH CONTROLLER CABLES & FORMATTED: \$425
- * ZUCKER COLOR CARD WITH PARALLEL PRINTER PORT: \$86
- * MONOCHROME/GRAPHICS VIDEO CARD WITH PRINTER PORT: \$89
- * PLEASE CALL FOR MONITOR PRICING

\$100 BUSS LIVES!
WE STILL SELL \$100 PRODUCTS AND C/P/M SINGLE BOARDS. FREE CATALOG.

SCHEMA: Schematic Capture and Drawing Package

WORKS WITH PC'S AND COMPATIBLES. EASY TO USE WITH OFF THE SHELF PRINTERS, Mouses, VIDEO CARDS, ETC. COMES COMPLETE WITH A HUGE COMPONENT DATABASE. SUPPORTS MONOCHROME, COLOR, OR EVEN EGA. FREE BROCHURE AND DEMO DISKETTE ON REQUEST. LIST \$500 — SPECIAL \$396.15

NEW! PC/XT EPROM PROGRAMMER \$199

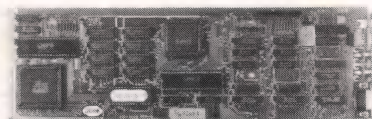


- * LATEST DESIGN * PROGRAMS UP TO 4 DEVICES AT ONE TIME * FEATURES EASY TO USE MENU DRIVEN SOFTWARE THAT RUNS UNDER PC OR MS-DOS.
- * USES AN INTELLIGENT PROGRAMMING ALGORITHM FOR SUPER FAST (8X) EPROM BURNING. * THIS PLUG-IN BOARD ATTACHES TO AN EXTERNAL MINI CHASSIS CONTAINING 4 TEXTOL Z.I.F. SOCKETS. * NO PERSONALITY MODULES REQUIRED * AUTOMATIC VPP SELECTION: 12.5V, 21V, OR 25V.
- * EPROM DATA CAN ALSO BE LOADED FROM OR SAVED TO A DISKETTE.
- * PROGRAMMING SOFTWARE SUPPORTS: 2716, 2732, 2732A, 2764, 2764A, 27128, 27128A, 27256, 27256A, 27512, AND 27512A. * ASSEMBLED AND TESTED, BURNED. IN WITH MANUAL. \$199 WITH SOFTWARE.

MICROTEK 2 M.B. EXPANDED MEMORY BOARD \$195 (O. K)
MEETS LOTUS/INTEL/MICROSOFT SPEC. FOR EXPANDED MEMORY. FOR USE AS EXPANDED MEMORY (EMS), ELECTRONIC DISK, OR PRINT SPOOLER. WITH A SUPERB MANUAL AND INSTALLATION SOFTWARE. FOR PC/XT OR MOST COMPATIBLES. FOR 2 MEGABYTES OF 150 NS RAM (72 DEVICES) INSTALLED AND BURNED IN ADD \$225.

PC/XT AT ENHANCED GRAPHICS ADAPTER

\$279



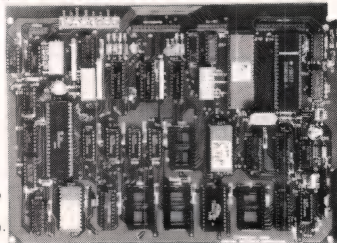
- * 3 CARDS IN ONE!
- * 100% IBM COMPATIBLE.
- * COMPATIBLE WITH IBM EGA, COLOR GRAPHICS ADAPTER, OR MONOCHROME ADAPTER.
- * 256K VIDEO RAM! (4 TIMES MORE THAN IBM!)
- * DUAL FREQUENCY OUTPUT FOR EITHER EGA OR STANDARD RGB COLOR MONITORS.
- * PERFECT MATE FOR NEC MULTISYNC COLOR MONITOR!
- * LIGHT PEN INPUT.
- * FULL 16 COLORS.
- * MFG. BY D.F.I., THE PREMIER FAR-EAST ADD-ON BOARD MAKER.
- * A SUPERIOR BOARD AT ABOUT 1/3 THE COST OF IBM EGA!
- * ASSEMBLED, TESTED, AND BURNED IN. WITH MANUAL.
- * USES CHIPS & TECH. VLSI 4 CHIP SET.
- * FIELD PROVEN BIOS.

ZRT-80 CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- * Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- * RS232 at 16 BAUD Rates from 75 to 19,200.
- * 24 x 80 standard format (60 Hz).
- * Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- * Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- * Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- * 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- * Composite or Split Video.
- * Any polarity of video or sync.
- * Inverse Video Capability.
- * Small Size: 6.5 x 9 inches.
- * Upper & lower case with descenders.
- * 7 x 9 Character Matrix.
- * Requires Par. ASCII keyboard.



\$8995 A&T
#ZRT-80 ADD
\$50
(COMPLETE KIT, 2K VIDEO RAM)

BLANK PCB WITH 2716
CHAR. ROM. 2732 MON. ROM

\$4995

SOURCE DISKETTE - ADD \$10
SET OF 2 CRYSTALS - ADD \$7.50

FOR 8 IN. SOURCE DISK
(C/P/M COMPATIBLE)
ADD \$10

THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD!

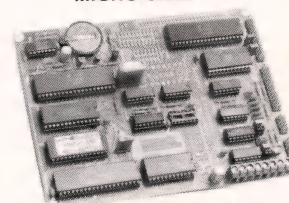
* FROM LINGER ENTERPRISES *

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. Use as a computer console or with a MODEM for hook up to any of the telephone-line computer services.

FEATURES:

- * Uses the new SMC 9028 Video Controller Chip coupled with a 6502A CPU.
- * RS-232 at 16 Baud Rates from 50 to 19,200
- * On board printer port!
- * 24 X 80 format (50/60 Hz).
- * For 15,750 Hz (Horiz.) monitors.
- * 3 Terminal Modes: H-19, ADM3A, and ANSI X 3.64-1979
- * Wide and thin-line graphics.
- * White characters on black background or reversed.
- * Character Attributes: De-inten, Inverse, Underline and Blank.
- * Low Power: 5VDC @ .7A, ± 12VDC @ 20MA.
- * Mini size: 6.5 X 5 inches.
- * Composite or split video.
- * 5 X 8 Dot Matrix characters (U/L case) with descenders.
- * Answer back capability.
- * Battery backed up status memory.
- * For ASCII parallel keyboard.

MICRO SIZE!



\$9995 (Full Kit)

ADD \$40 FOR A&T

SOURCE DISKETTE:
PC/XT FORMAT
5 1/4 IN. \$15

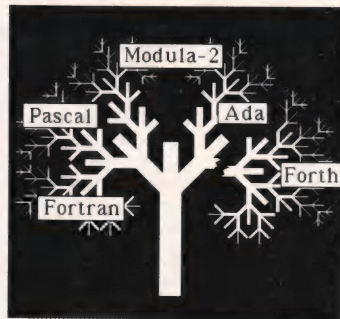
NEW OPTION
PC/XT keyboard ROM. Allows use of IBM style keyboard. \$15

TERMS: Add \$3.00 postage. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance. Prices subject to change without notice.

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

Factoring in Forth



CREATE... DOES> is the pearl of Forth, a way to wrest control from the compiler and vest it in the programmer, where Forth programmers believe it rightfully belongs. But CREATE... DOES> is not just a power play, a blow struck for programmer independence; it is also an example of superb factoring. In this column, I want to talk about factoring, an elusive thread woven throughout the fabric of Forth. Because it is so elusive, sneaking up on it metaphorically might be the best approach.

You first find factoring when you create commands. Forth programmers routinely create new commands; that is how Forth programs are written. Many Forth programmers arrive from other languages and are familiar with procedures and named subroutines. Their habits and expectations from that prior experience lead them astray. Instead of short commands, they write large chunks of code, difficult to debug and fitting only the particular situation that prompted them.

Their former languages required separate compilation and linking, and the economics of that overhead made it sensible to pack procedures with enough code to balance the time and effort to compile and test it. In Forth, though, the totally interactive compiler is right at hand, and each word is a self-contained module that can be run alone. Compilation is cheap, linking is less, so the module can be small—in

by Michael Ham

fact, it should be small.

Veteran Forth programmers smile tolerantly at the novice's monster word and in its place produce two dozen tiny words. These words are typically bug-free, each being utterly simple, and they snap together like

Leggo blocks to build a command that efficiently accomplishes the same task as the beginner's awkward monster. Even better, the little words can be assembled in many ways and thus find many uses.

The veteran builds general-purpose tools from the elements of the solution. The beginner, unable to locate the separate essences, constructs a word that addresses the entire compound situation. The veteran's skill at factoring consists of being able to find the independent components implicit in the task and to define those words first.

To take a simple example: Suppose the programmer needs to calculate a percentage, rounded to tenths, and display it at a certain location on the screen. Listing One, page 94, shows how a beginner might do this: The word % does everything required—that is, too much. The veteran automatically factors the requirement into several commands, as shown in Listing Two, page 94. One word calculates the percentage to tenths, leaving the result on the stack. Another formats an ASCII string that represents the number found on the stack, assuming the least significant digit represents tenths. These two words are then used to define a word that calculates the percentage, formats the string, and then displays it. Cursor placement is handled separately so that this word can be used for a display anywhere on the screen—or on a printer, for that matter. In fact, in addition to the general-purpose percentage-display word, we now have a word that calculates percentages and a word that shows a number as

tenths. Other words can be defined from these tools—for example, the word % .R displays the percentage flush right in a field of specified width.

But factoring involves more than writing a series of small definitions that fit together to address a task. It requires finding the “true” divisions, teasing apart the whole to reveal its internal structure. To factor properly, find the subtasks nestled within the task. Factoring requires a sensitivity to the underlying structure of the situation. If the problem itself is elementary, to find its parts is of course no problem. For programmers, however, problems arrive entangled in each other, embedded in assumptions and past practice, and often not even announced as problems. The first hint that the factoring is bad may be that the definition is difficult to write.

The rightness of the factoring is marked by a simplicity, but that simplicity is not easily attained and is also somewhat deceptive. To call a definition simple just because it contains few commands is a triumph of synthesis. Our ability to chunk knowledge and convert a name from pointer to entity enables us to hide intricacies beneath the skin of a single concept. From complexity we can extract simplicity.

Our minds want to find or make a unity. Factoring fights upstream against this tendency. Factoring requires us to forsake unity and probe the problem to locate the separate masses that form the unit: locate them, separate them, and name them.

Good analysis teases apart the whole, finding and revealing the true divisions. It is done less by logic than by an inward tactile sense that can somehow distinguish masses still hidden in the dark of ignorance. Once we grasp the inner structure and break the problem down, the divisions stand exposed to the illumina-

Product Information

Free!

Dr. Dobb's Journal of Software Tools

OCTOBER 1986 #120 Expiration Date: JANUARY 31, 1987

Name _____ Title _____
 Company _____ Phone _____
 Address _____
 City/State/Zip _____

Please circle one letter in each category:

I. My work is performed:

- A. for in-house use only.
- B. for other companies.
- C. for end users/retailers.
- D. in none of the above areas.

II. My primary job function:

- A. Software Project Mgmt./Spvr
- B. Hardware Project Mgmt./Spvr
- C. Computer Consultant
- D. Corporate Management
- E. Other

III. My company department performs:

- A. software development.
- B. computer system integration.
- C. computer manufacturing.
- D. computer consulting.
- E. computer research.
- F. none of the above.

IV. This inquiry is for:

- A. a purchase within 1 month.
- B. a purchase within 1 to 6 months.
- C. product information only.

V. Corporate Purchase Authority:

- A. Final Decision-maker
- B. Approve/Recommend
- C. No Influence

VI. Personal Computer Users at my Jobsite:

- A. 10,000 or more
- B. 500 to 9,999
- C. 100 to 499
- D. 10 to 99
- E. less than 10

VII. On average, I advise others about computers:

- A. more than once per day.
- B. once per day.
- C. once per week.
- D. less than once per week.

VIII. In my job function, I:

- A. design software and/or write code.
- B. design software.
- C. write code.
- D. don't design software or write code.

A Reader Service number appears on each advertisement. Circle the corresponding numbers below for more info.

001	002	003	004	005	006	007	008	009
010	011	012	013	014	015	016	017	018
019	020	021	022	023	024	025	026	027
028	029	030	031	032	033	034	035	036
037	038	039	040	041	042	043	044	045
046	047	048	049	050	051	052	053	054
055	056	057	058	059	060	061	062	063
064	065	066	067	068	069	070	071	072
073	074	075	076	077	078	079	080	081
082	083	084	085	086	087	088	089	090
091	092	093	094	095	096	097	098	099
100	101	102	103	104	105	106	107	108
109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153
154	155	156	157	158	159	160	161	162
163	164	165	166	167	168	169	170	171
172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189
190	191	192	193	194	195	196	197	198
199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216
217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234
235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252
253	254	255	256	257	258	259	260	261
262	263	264	265	266	267	268	269	270
271	272	273	274	275	276	277	278	279
280	281	282	283	284	285	286	287	288
289	290	291	292	293	294	295	296	297
298	299	999						

Circle 999 to start a 12 month subscription at the price of \$29.97

Postage Paid!

Thank You!
 Dr. Dobb's greatly appreciates your responses to questions I through VIII.

Product Information

Free!



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

First Class Permit #217, Clinton, Iowa

POSTAGE WILL BE PAID BY ADDRESSEE

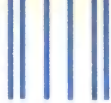
Dr. Dobb's Journal of Software Tools

FOR THE PROFESSIONAL PROGRAMMER

P.O. Box 2157

Clinton, Iowa 52735-2157





NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

First Class Permit #217, Clinton, Iowa

POSTAGE WILL BE PAID BY ADDRESSEE

**Dr. Dobb's Journal of
Software Tools**
FOR THE PROFESSIONAL PROGRAMMER
P.O. Box 2157
Clinton, Iowa 52735-2157



**Product
Information**

Free!

Thank You!

**Dr. Dobb's greatly appreciates your
responses to questions I through VIII.**

Postage Paid!

A Reader Service number appears on each advertisement. Circle the corresponding numbers below for more info.

001 002 003 004 005 006 007 008 009
010 011 012 013 014 015 016 017 018
019 020 021 022 023 024 025 026 027
028 029 030 031 032 033 034 035 036
037 038 039 040 041 042 043 044 045
046 047 048 049 050 051 052 053 054
055 056 057 058 059 060 061 062 063
064 065 066 067 068 069 070 071 072
073 074 075 076 077 078 079 080 081
082 083 084 085 086 087 088 089 090
091 092 093 094 095 096 097 098 099
100 101 102 103 104 105 106 107 108
109 110 111 112 113 114 115 116 117
118 119 120 121 122 123 124 125 126
127 128 129 130 131 132 133 134 135
136 137 138 139 140 141 142 143 144
145 146 147 148 149 150 151 152 153
154 155 156 157 158 159 160 161 162
163 164 165 166 167 168 169 170 171
172 173 174 175 176 177 178 179 180
181 182 183 184 185 186 187 188 189
190 191 192 193 194 195 196 197 198
199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216
217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234
235 236 237 238 239 240 241 242 243
244 245 246 247 248 249 250 251 252
253 254 255 256 257 258 259 260 261
262 263 264 265 266 267 268 269 270
271 272 273 274 275 276 277 278 279
280 281 282 283 284 285 286 287 288
289 290 291 292 293 294 295 296 297
298 299 999

Circle 999 to start a 12 month subscription at the price of \$29.97

Dr. Dobb's Journal of Software Tools

OCTOBER 1986 #120 Expiration Date: JANUARY 31, 1987

Name _____ Title _____

Company _____ Phone _____

Address _____

City / State / Zip _____

Please circle one letter in each category:

I. My work is performed:

- A. for in-house use only.
- B. for other companies.
- C. for end users/retailers.
- D. in none of the above areas.

V. Corporate Purchase Authority:

- A. Final Decision-maker
- B. Approve/Recommend
- C. No Influence

II. My primary job function:

- A. Software Project Mgmt/Spvr
- B. Hardware Project Mgmt/Spvr
- C. Computer Consultant
- D. Corporate Management
- E. Other

VI. Personal Computer Users at my Jobsite:

- A. 10,000 or more
- B. 500 to 9,999
- C. 100 to 499
- D. 10 to 99
- E. less than 10

III. My company department performs:

- A. software development.
- B. computer system integration.
- C. computer manufacturing.
- D. computer consulting.
- E. computer research.
- F. none of the above.

VII. On average, I advise others about computers:

- A. more than once per day.
- B. once per day.
- C. once per week.
- D. less than once per week.

IV. This inquiry is for:

- A. a purchase within 1 month.
- B. a purchase within 1 to 6 months.
- C. product information only.

VIII. In my job function, I:

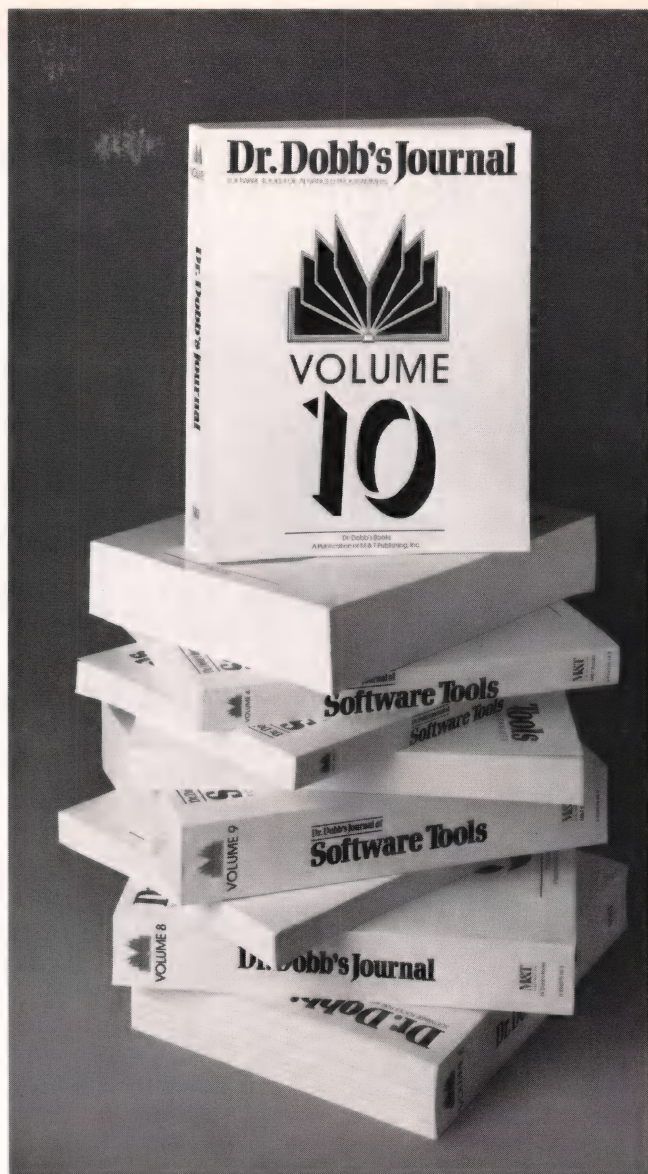
- A. design software and/or write code.
- B. design software.
- C. write code.
- D. don't design software or write code.

**Product
Information**

Free!

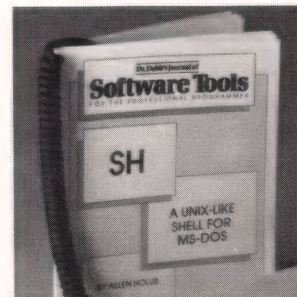
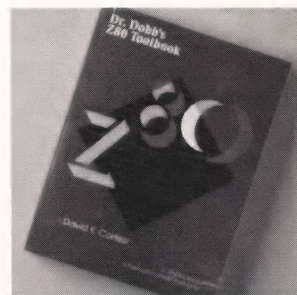
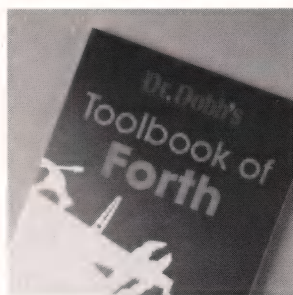
DR. DOBB'S CATALOG

featuring:
BOUND VOLUME
10



ALSO INSIDE:

A UNIX-LIKE SHELL &
UTILITY PACKAGE FOR MS-DOS
DR. DOBB'S COMPLETE C TOOLBOX
DR. DOBB'S LISTINGS ON DISK
THE TOOLBOOK OF FORTH
Z80 TOOLBOOK



DR. DOBB'S BOUND VOLUME 10



JUST RELEASED!

**INCLUDES ALL OF 1985:
THE YEAR DDJ SOLD OUT OF ALMOST
EVERY BACK ISSUE! NOW, THE
ENTIRE EDITORIAL CONTENTS
OF 1985 IS YOURS, IN
ONE GIANT VOLUME.**

Bound Volume #10: The year of living dangerously. In 1985, iconoclastic DDJ beat Apple to the goal of adding more memory, a SCSI port, and a hard disk to the Macintosh. We dared to criticize the much-praised Turbo Pascal, challenged the Unix establishment with plans for a free Unix, and asked hard questions about privacy and control in the Information Age. Of course we also kept the technical level high, with the most exhaustive review ever of programmers' editors and of C compilers, and with powerful software tools in C, Modula-2, Forth, Pascal, assembly language, and Prolog.
Item #020D \$35.75

SAVE 20% WHEN YOU ORDER THE COMPLETE 10 VOLUME SET!

Receive the 10 volumes of valuable code and commentary that helped shape the computer industry, for over \$65 off! That includes the entire editorial contents of Dr. Dobb's from 1976 through 1985 for only \$262. You'll find this set practically useful and historically fascinating!

Item #020E \$262

TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT

A DECADE OF SOFTWARE TOOLS

Bound Volume #1: 1976 The working notes of a technological revolution. Before there was an Apple, DDJ put a programming language on the first microcomputers, and became chronicler and instrument of the microcomputer revolution.

Item #013 \$30.75

Bound Volume #2: 1977 Running light without overbyte. By year two the formula was clear: serious technical questions handled with a minimum of reverence; much source code; and a commitment to tight coding.

Item #014 \$30.75

Bound Volume #3: 1978 The roots of Silicon Valley growth. The S-100 bus was hashed out in DDJ's pages. Steve Wozniak and others published in DDJ code that would help build an industry.

Item #015 \$30.75

Bound Volume #4: 1979 In the midst of the gold rush. Three years before IBM moved in, the neighborhood was less civilized. DDJ published a gold mine of tips, tricks, and algorithms.

Item #016 \$30.75

Bound Volume #5: 1980 C and CP/M. 1980 saw an all-CP/M issue, including Gary Kildall's history of CP/M, and Ron Cain's original Small-C compiler.

Item #017 \$30.75

Bound Volume #6: 1981 The First of Forth. This was the year DDJ launched its first Forth issue and Dr. Dobb's Clinic. Plus: PCNET, the Conference Tree, and 6809 Tiny BASIC.

Item #018 \$30.75

Bound Volume #7: 1982 Legitimacy. DDJ observed the IBM phenomenon, reviewed MS-DOS and CP/M-86, and looked forward to fifth-generation computers.

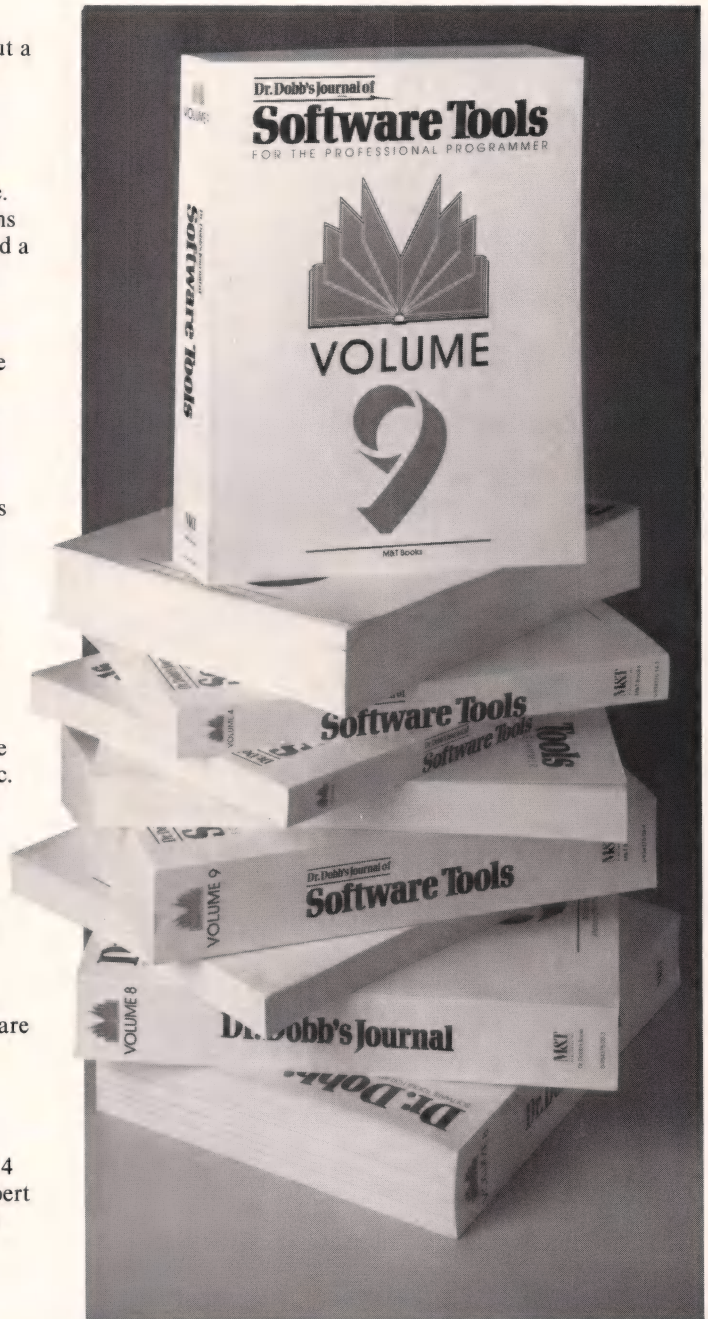
Item #019 \$35.75

Bound Volume #8: 1983 Power tools. Professional software development on a PC was getting easier; DDJ helped, with Small-C, the RED editor, and an Ada subset.

Item #020 \$35.75

Bound Volume #9: 1984 Shaping things to come. In 1984 DDJ examined new programming environments: Prolog, expert systems, Modula-2, and a \$49.95 Pascal. Plus Allen Holub's GREP, Unix internals, and two encryption systems.

Item #020B \$35.75



SAVE 15%

WHEN YOU ORDER 4 OR MORE VOLUMES

Choose any 4 DDJ Bound Volumes and take 15% off the total price!

TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
 AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT

DR. DOBB'S C TOOLBOX: POWERFUL

DR. DOBB'S C SOFTWARE TOOLS ON DISK

To complement the *Toolbook*, *Dr. Dobb's* offers the following programs on disk. Documentation and full C source code are included.

Except where indicated, both CP/M and MS/PC-DOS versions are available.

SMALL-C COMPILER

Jim Hendrix's Small-C Compiler is the most popular piece of software published in *Dr. Dobb's* 11-year history. Like a home-study course in compiler design, the Small-C Compiler and the *Small-C Handbook* provide everything you need but the computer for learning how compilers are constructed, and for learning C at its most fundamental level.

Available for MS/PC-DOS or CP/M systems. Please specify format.

Small-C Compiler Item #007 \$19.95

SMALL-MAC: AN ASSEMBLER FOR SMALL-C

This assembler features simplicity, portability, adaptability, and educational value. The package includes:

- a simplified macro facility
- C language expression operators
- object file visibility
- descriptive error messages
- an externally defined instruction table

You get the macro assembler, linkage editor, load-and-go loader, library manager, CPU configuration utility, and a utility to dump relocatable files. Documentation is also included. For CP/M systems only. Please specify format.

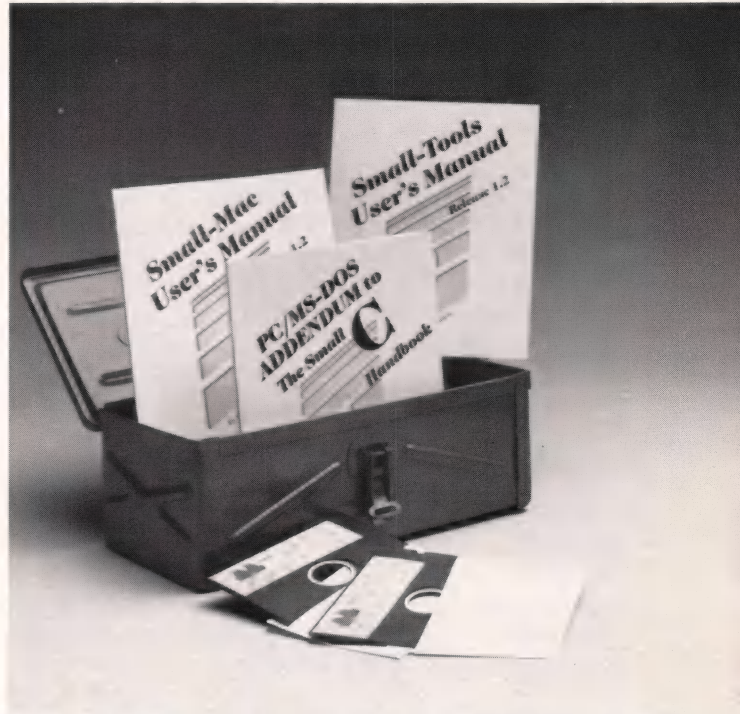
Small-Mac Item #012A \$29.95

SMALL-TOOLS: PROGRAMS FOR TEXT PROCESSING

This package of programs performs specific, modular operations on text files, including: editing; formatting; sorting; merging; listing; printing; searching; changing; transliterating; copying; concatenating; encrypting and decrypting; replacing spaces with tabs and tabs with spaces; counting characters, words, or lines; and selecting printer fonts.

Small-Tools is supplied in source code form so you can select and adapt these tools to your own purposes. Documentation is also included.

Small-Tools Item #010A \$29.95



DR. DOBB'S TOOLBOOK OF C

Over 700 pages of C material, including articles by such C experts as Kernighan and Ritchie, Cain and Hendrix, Skjellum and Holub! The level is sophisticated and pragmatic, appropriate for professional C programmers.

The most valuable part of the *Toolbook* to many will be the hundreds of pages of useful C source code, including:

- Jim Hendrix's famous Small-C Compiler and New Library for Small C—Also available on disk!
- NEW! Hendrix's Small Mac: An Assembler for Small C and Small Tools: Programs for Text Processing—Both also available on disk!
- All of Anthony Skjellum's C Programmer's Notebook columns distilled by Tony into one thought-provoking chapter

The accompanying text explains, in the programmers own words, why they did what they did.

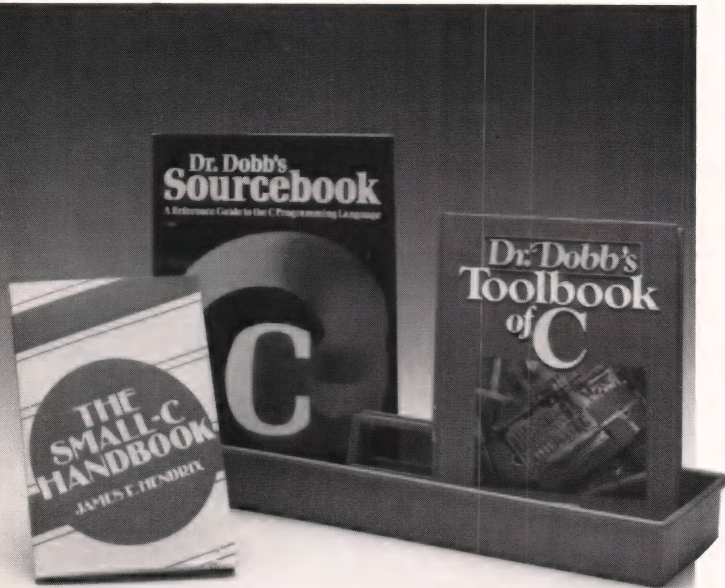
You'll find all the best C articles and code published in *Dr. Dobb's* over the years updated for the *Toolbook*, including Ron Cain's original Small-C article.

From M&T Publishing and Brady Communications

Dr. Dobb's Toolbook of C Item #005 \$29.95

TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT

TOOLS FOR SOFTWARE DEVELOPERS



DR. DOBB'S SOURCEBOOK:

THE C PROGRAMMING LANGUAGE REFERENCE

A comprehensive reference manual to new information, products and services, the Sourcebook contains:

- a bibliography of over 300 articles and books on C
- a descriptive list of products for C programmers, including compilers, editors, interpreters, and utilities
- a list of C-related services: classes, seminars, and on-line services

Dr. Dobb's Sourcebook

Item #004 \$7.95

SPECIAL PACKAGES 20% OFF

Receive a complete set of Dr. Dobb's C programming tools for your MS/PC-DOS or CP/M system for 20 percent off the combined individual prices!

CP/M C PACKAGE

Receive this special package and save \$20! You'll get:

- Dr. Dobb's Toolbook for C
- The Small-C Handbook
- The Small-C Compiler on disk
- The Small-Mac assembler on disk, with documentation
- The Small-Tools text-processing programs on disk, with documentation all for only \$99.95!

Please specify format.

CP/M C Package

Item #005A \$99.95

MS/PC-DOS C PACKAGE

Save \$20 when you order this special package. You'll receive:

- Dr. Dobb's Toolbook of C
- The Small-C Handbook with the MS/PC DOS Addendum
- The Small-C Compiler on disk
- The Small Tools text-processing programs on disk, with manual all for only \$82.95

MS/PC-DOS C Package

Item #005B \$82.95

THE SMALL-C HANDBOOK

Jim Hendrix's *Small-C Handbook* is the reference book on his Small-C Compiler. In addition to describing the operation of the compiler, the book contains complete source listings to the compiler and its library of arithmetic and logical routines. A perfect companion to the Hendrix Small-C Compiler available from DDJ on disk, the *Handbook* even tells you how to use the compiler to generate a new version of itself! While both the *Handbook* and the *Toolbook* provide documentation for the Small-C Compiler, the *Handbook* contains a more detailed discussion and is available with addendum for the MS/PC-DOS version.

From M&T Publishing and Brady Communications

The Small-C Handbook Item #006 \$17.95

The Handbook with MS/PC-DOS Addendum Item #006A \$22.95

C DISK FORMATS

When ordering, please indicate MS/PC DOS or CP/M. For CP/M disks, please specify one of the following formats: Apple, Osborne, Kaypro, Zenith Z-100 DS/DD, 8" SS/SD. Special order formats are available for an additional \$10 each.

TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT

DR. DOBB'S 1986 LISTINGS ON DISK

Dr. Dobb's Journal of Software Tools offers the convenience of selected listings on disk.

DR. DOBB'S LISTINGS #1

JANUARY-APRIL 1986

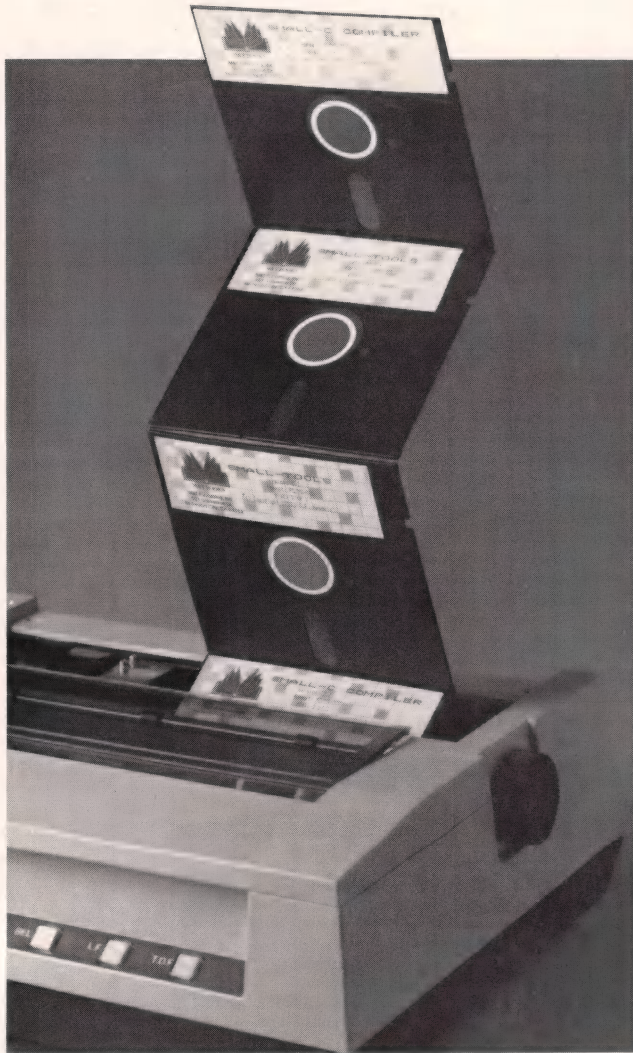
Includes listings from the following articles, and more.

January Issue #111

"A Simple OS for Realtime Applications; 68000 Assembly Language Techniques for an Operating System Kernel" by *DDJ* editor Nick Turner.

"Exec Calls and FORTRAN; A Technique Allowing Execution of User or System Task from a User Program" from *DDJ's* 16-Bit Software Toolbox, by Robert Sypek.

"32-bit Square Roots; An 8086 Assembly Language Routine for 32-bit Square Roots" by Michael Barr.



February Issue #112

"Fast Integer Powers for Pascal; An Implementation of the Fastest-Known Algorithm for the Computation of Integer Powers" by Dennis E. Hamilton.

"Data Abstraction with Modula-2" by Bill Walker and Stephen Alexander.

"Learning Ada on a Micro; A Draw Poker Program in Ada" by DoWhile Jones.

"Fast IBM PC Graphics Routines" from *DDJ's* 16-Bit Software Toolbox, by Dan Rollins.

March Issue #113

"Recursive Bose-Nelson Sort; An Alternative to Joe Celko's September 1985 Sort Routine" by R. J. Wissbaum.

"A Variable-Metric Minimizer; A C Program for Minimizing Arbitrary Functions" by Joe Marasco.

"Concurrency and Turbo Pascal; An Approach to Implementing Coroutines in Pascal" by Ernest Bergmann.

"Speeding MS/DOS Disk Access; Programs to Test Disk Access Speed" by Greg Weissman.

"Square Roots on the NS32000; Comparable Square Root Routines in C and Assembly Language for National Semiconductor's 32000 Family" by Richard Campbell.

April Issue #114

"Boca Raton Inference Engine; Lisp, Prolog, and Expert-2 Techniques and Code" by Robert Brown.

Dr. Dobb's Listings #1/86

Item #170 \$14.95

DR. DOBB'S LISTINGS #2

MAY-AUGUST 1986

Includes listings from the following articles, and more.

May Issue #115

"Simple Plots with the Enhanced Graphics Adapter" by Nabajyoti Barkakati.

"The Cryptographer's Toolbox" by Fred A. Scacchitti.

June Issue #116

"Structured Programming; Overloading Procedures, Exporting Opaque Types, Data Hiding" by Namir Shamas.

"Compuserve B Protocol" by Steve Wilhite.

July Issue #117

"Structured Programming; Tiny Tools, Array-Defining Words" by Michael Ham.

August Issue #118

"Structured Programming; Generic Routines in Ada and Modula-2, Extended For Loop" by Namir Shamas.

Dr. Dobb's Listings #2/86

Item #171 \$14.95

Please specify MS-DOS, Macintosh, or CP/M.

For CP/M disks specify: Apple, Osborne, Kaypro, Zenith

TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
 AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT

THE DR. DOBB'S TOOLBOOK SHELF

Z80 TOOLBOOK

David E. Cortesi, longtime Dr. Dobb's columnist brings you:

- **a method of designing programs and coding them in assembly language.** Cortesi walks through the initial specifications, designing an algorithm and writing the code. He demonstrates the construction of several useful programs.
- **a complete, integrated toolkit of subroutines** for arithmetic, for string-handling, and for total control of the CP/M file system. They bring the ease and power of a compiler's run-time library to your assembly language work, without a compiler's size and sluggish code.
- **Every line of the toolkit's source code is there to read.**

ORDER THE Z80 SOFTWARE ON DISK!

All the software in Dr. Dobb's Z80 Toolbook—the programs plus the entire toolkit, both as source code and object modules for both CP/M 2.2 and CP/M Plus—is yours on disk! Most of the programs are included in the book, however, the disk is necessary for complete listings. A Z80 microprocessor and a Digital Research International RMAC assembler or equivalent are required.

Dr. Dobb's Z80 Toolbook	Item #022	\$25
Dr. Dobb's Z80 Toolbook w/disk	Item #022A	\$40

Please specify one of the following disk formats: 8" SS/SD, Apple, Osborne, or Kaypro

DR. DOBB'S TOOLBOOK OF FORTH

This comprehensive collection of useful Forth programs and tutorials contains DDJ's best Forth articles, expanded and revised along with new material. In addition, you'll glean important insights about the potential of this increasingly popular language from the many in-depth discussions of advanced Forth topics. You'll find sections on:

Mathematics in Forth, including "Series Expansion in Forth," "Forth Floating-Point Package," and "Signed Integer Division"

Modifications/Extensions, including "A Proposal for Strings in Forth," "Non-Deterministic Control Words," "Some Forth Coding Standards," and "Towards a More Writable Forth Syntax"

Forth Programs, including "GO in Forth," "Elements of a Forth Data-Base Design," "The Forth Sort," "SEND & RECV," "Interface for a Mouse," "Relocating Loader in Forth," "Forth Decompiler," "Screen-Oriented Editor ReVisited," "Evolution of a Video Editor," "H-19 Screen Editor," and "The Conference Tree."

Forth—the language, including "The Forth Philosophy," "Teaching Forth as a First Language," and "Forth-83 and Vocabularies"

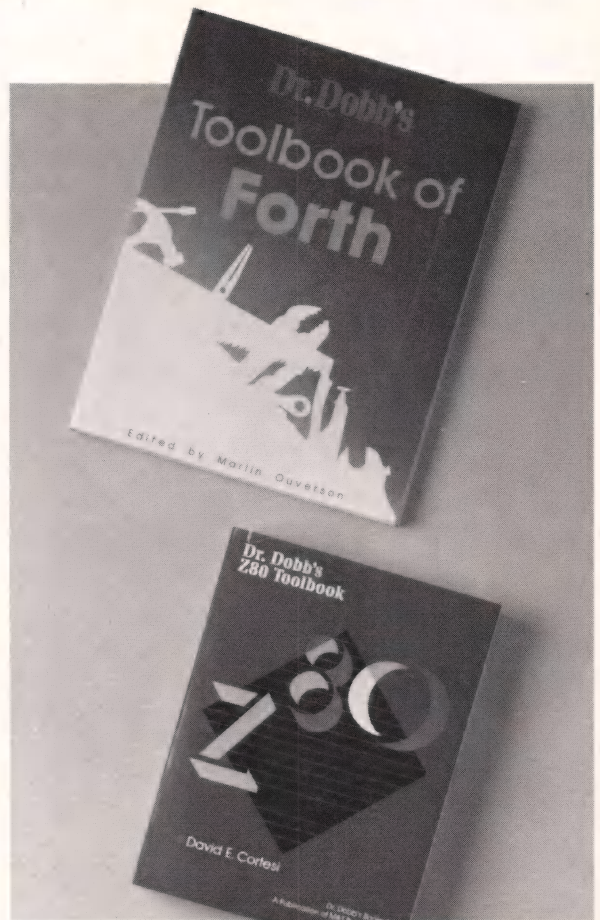
Implementing Forth, including "Forth and the Motorola 68000," "A 68000 Forth Assembler," "A Forth Assembler for the 6502," and "Z8000 Forth." You'll also find Appendices that will help you convert fig-Forth to Forth-83, and tell you how to stay up-to-date on the latest developments and refinements of this popular language.

The screens in the book are also available on disk as ASCII files. Receive Dr. Dobb's Toolbook of Forth, along with the software on disk, together for only \$39.95.

Dr. Dobb's Toolbook of Forth	Item #030	\$22.95
Dr. Dobb's Toolbook of Forth w/Disk	Item #031	\$39.95

Please specify MS/PC-DOS, Apple II, Macintosh, or CP/M. For CP/M disks, specify Osborne or 8" SS/SD.

TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
 AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT



SH: A UNIX-LIKE SHELL FOR MS-DOS

Includes complete source code and documentation

THE SHELL V.2 NEW, UPGRADED VERSION!

The shell is an MS-DOS implementation of the most often used parts of the UNIX C Shell. This package includes an executable version of the Shell, along with **complete source code and full documentation**, by DDJ columnist Allen Holub. If you are a registered user, or have already purchased Version 1 of the Shell from Dr. Dobb's, you can receive the upgrade disk for only \$6. Supported features include:

Editing Command-line editing with the cursors is supported. The line is visible as you edit it.

Aliases Can be used to change the names of commands or as very fast, memory-resident, batch files. Nested aliases are supported.

History You can execute previous commands. The command can be edited before being executed. Version 2 supports imbedded history requests (Bar; !! >foo).

Redirection and Pipes <> >> >& >>& !

Pipe temporary files can be put on a RAM disk.

Unix-like Command Syntax / can be used to separate directory names (\ can now be used as well). A 2048-byte command line is supported. Command-line wild card expansion. Multiple commands on a line.

DOS-compatible prompt support

\$d \$t \$_ \$e \$h \$n \$q \$\$ \$%

C-Shell Based Shell Scripts (batch files) Shell Variables are macros that can be used on the command line. Version 2 supports arithmetic manipulation of shell variables using the @ command. The following C operators are supported: () + - * / % <= >= <> != == ! && !! =

A batch file can call another batch file like a subroutine. Control is passed to the second file and then back to the first when the second is finished. Batch files can return values to the calling file using the exit and \$status mechanisms.

A powerful, interpretive, programming language, based on the UNIX C Shell, is now supported, including:

if/then/else	foreach	break
while	switch/case	continue

All commands can be nested.

The shell runs on IBM PC's and compatibles.

The Shell	Item #160	\$29.95
------------------	------------------	----------------

The Shell Upgrade Disk	Item #160A	\$6.00
<i>(for owners of the Shell Version 1)</i>		

**SAVE OVER 15%! ORDER THE SHELL
AND /UTIL TOGETHER FOR ONLY \$50!**

Shell /Util Package	Item #162	\$50
----------------------------	------------------	-------------

/UTIL

/Util is a collection of UNIX-like utility programs for MS-DOS. This package includes updates of the highly acclaimed Dr. Dobb's articles; Grep: a UNIX-like Generalized Regular Expression Processor, and LS and Getargs from DDJ's C Chest.

Source code is included and all programs (and most of the utility subroutines) are fully documented in a UNIX-style manual. You'll find executable versions of:

cat	echo	mv	rm
cp	grep	p	rmdir
date	Ls	pause	sub
du	mkdir	printenv	chmod

/Util	Item #161	\$29.95
--------------	------------------	----------------

**TO ORDER: RETURN THE FORM AT THE END OF THE CATALOG, OR
CALL TOLL-FREE 1-800-528-6050 EXT 4001
AND REFER TO PRODUCT ITEM NUMBER, TITLE AND DISK FORMAT**

ORDER FORM

ORDER NOW!

NAME _____

(Please use street address, not P.O. Box)

ADDRESS _____

CITY _____

STATE _____

ZIP _____

DAY PHONE _____

For disk orders, please indicate format. Refer to ad for standard format availability for each product. Special formats available for additional \$10 each.

☐ MS/DOS

☐ CP/M

_____ Zenith Z-100 DS/DD

☐ Macintosh

_____ Kaypro

_____ Osborne

☐ Apple II

_____ 8" SS/SD

_____ Apple

For Faster Service
On Credit Card Orders

CALL TOLL FREE
1-800-528-6050
Ext. 4001

Please Refer To Item #
When Ordering



QUANTITY	ITEM #	DESCRIPTION	UNIT PRICE	TOTAL PRICE

SUB-TOTAL _____

CA residents must add applicable sale tax on merchandise total _____ % _____

SALES TAX _____

(CA residents must add sales tax EXCEPT *Dr. Dobb's Sourcebook* #004)

Shipping must be included with order. See rates below. _____

SHIPPING _____

TOTAL ORDER _____

☐ VISA

NAME ON CARD _____

☐ MASTERCARD

ACCOUNT NO. _____

☐ AMERICAN EXPRESS

EXPIRATION DATE _____

☐ CHECK (make checks payable
to M&T Publishing)

SIGNATURE _____

In U.S. For Bound Volumes, add \$2.25 per book. Add \$8.75 for special C Packages. For other books and disks, add \$1.75 per item.

Outside U.S. For Bound Volumes, add \$5.25 per book surface mail. Add \$18 surface mail for Special C Packages. For other books and disks, add \$3.25 per item surface mail. Foreign airmail rates available on request.

PROMPT DELIVERY! DEALER INQUIRY WELCOME!

DR. DOBB'S CATALOG ORDER FORM

Please fold along fold-line and staple or tape closed.



No Postage
Necessary
If Mailed
In The
United States

BUSINESS REPLY MAIL

First Class Permit No. 790 Redwood City, CA

Postage Will Be Paid By Addressee

DR. DOBB'S CATALOG

501 GALVESTON DRIVE
REDWOOD CITY, CA 94063

Please fold along fold-line and staple or tape closed.

STRUCTURED PROGRAMMING (continued from page 104)

tion of understanding. Anyone can see them once the job is done. But when the problem is still murky—perhaps not even yet seen as a problem—recognizing its separate parts is difficult. And if the analysis is done humpty-dumpty, the pieces won't fit together again. Factoring thus requires us to understand which functions belong together and which do not. Functions are factored together or apart depending on how they fit in the overall structure. In both directions, names are the navigation lights. We recognize (create?) a unity by assigning a name to a group; we analyze a unity by naming its parts. Names are our guide and our tool.

Analysis is very good indeed at marking the path once a good factoring has been reached, but the factoring itself may have been achieved directly through an alert awareness toward your experience with the problem. You live and work with the problem for days or even weeks, then "suddenly" the solution is obvious. Analysis then discovers or constructs the reasons this approach is sound.

For example, I wrote a program recently in which the user enters the date. To avoid possible ambiguity, I labeled separate fields for the month, day, and year. The user can move from field to field with the arrow keys, and when he or she types an entry and presses Enter, the cursor moves to the next field. For weeks I unconsciously first pressed / (associated in my mind with entering the date) and then, when / had no effect, automatically pressed Enter. One day I noticed what I was doing. Once I had noticed, the solution was simple: I made / (and for good measure — as well) equivalent to Enter in the date-entry routine.

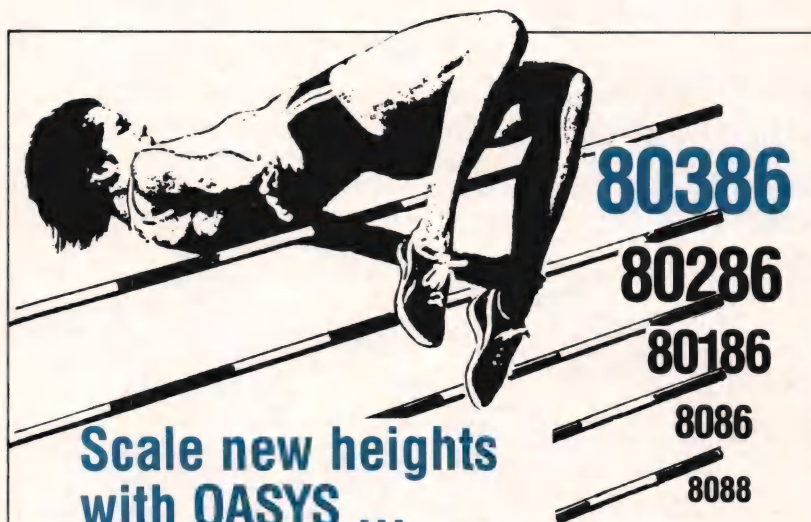
Another example of the slow surfacing of a buried problem occurred in the same routine. The routine is smart enough to know that months get no larger than 12 and days no larger than the maximum for the current month. It observes these limits when you type new digits, which enter the display from the right. If you enter 1 for a day and then type 7, for example, the display shows 17; typing the

same sequence for a month leaves the display with 7, not 17.

With this routine, you can usually correct a typo just by typing the correct number: If you type 7 instead of 8, you can correct it simply by typing 8—78 will not appear for a month or a day. Sometimes, however, you have to repeat a key to get it to "take"—for the month, if you type 1 when you mean 2, following the 1 by 2 results in 12 (still not what you mean), but typing a second 2 produces 2 (not 22, which is invalid for a month). The same pattern works for

the day. Suppose you type 2 instead of 5. If you now type 5, you get 25, but typing another 5 produces the 5 you want because 55 is invalid for a day.

Repeating the number feels familiar, like repeating something to an inattentive listener. It fails, however, to work with 11 and with 22 (for day). If you type 1 when you mean 2, for example, typing the 2 produces 12, a second 2 produces 22, and all subsequent 2s leave the number 22. It took me a long time to become aware that 1 and 2 didn't work in the same way as did the other numbers, even



Scale new heights with OASYS ...

Our Intel Software Tool Kit keeps on growing ...

TOOL KIT

- Fully optimizing C, Pascal and FORTRAN Green Hills Compilers with full 80386 Support
- Wizard C for 8086/186/286
- 80386 Phar Lap Macro Assembler
- 80386 Phar Lap Linker with Intel and COFF object formats
- Symbolic Debuggers
- Full complement of compilers, assemblers, linkers, debuggers and simulators for 8086/186/286

FEATURES

- 80386 compilers utilize multiple optimization techniques producing extremely compact code
- Tool Kit provides complete support for 80387 and Weitek 1167 Floating Point Units
- Complete Run-Time Library support
- 80386 Assembler accepts Intel/Microsoft instruction mnemonics
- Produces COFF and Intel-compatible Object modules
- Assembler/Linker backwards compatible with entire 8086 family, including 8086/80186/80286
- Softprobe II 8086/186 Simulator and Debugger
- Wizard C compiler for 8086/186/286 including 286 "Protected Mode"
- Full UNIX support

AVAILABILITY

Native on 80386
For Cross-Development Targeting
80386/286/186/86:
VAX; Sun; IBM PC; OASYS PC Platform™
PC Co-Processor Board; ...Others being ported

You name it ...

We provide a "One-Stop Shopping" service for more than 100 products running on, and/or targeting to, the most popular 32-, 16- and 8-bit micros and operating systems.

We Specialize In:

Cross/Native Compilers C, Pascal, FORTRAN, Ada, LISP — Assemblers/Linkers — Symbolic Debuggers — Simulators — Interpreters — Profilers — QA Tools — Design Tools — Comm. Tools — OS Kernels — Editors — VAX & PC — Attached Processors and more

We Support:

680xx, 80x86, 320xx, 68xx, 80xx, Clipper, and dozens more

We Provide:

Porting and technical assistance — OEM arrangements — Custom development — Let us be your external tools group

A DIVISION OF XEL

Oasys

60 Aberdeen Avenue, Cambridge, MA 02138 (617) 491-4180

Trademarks are acknowledged to: Intel Corp., AT&T, DEC, U.S. Government (AJPO), Weitek Corp., Phar Lap Software Systems & Software Inc., Wizard Systems Software, and XEL, Inc.

Circle no. 254 on reader service card.

STRUCTURED PROGRAMMING

(continued from page 113)

though I did experience their behavior. I was, however, vaguely aware that something was wrong, and when I finally realized what it was, I easily changed the program so that it would realize that 1 typed when 11 was present did not mean 11—11 was already there. In that case the program drops 11 and leaves only 1. Similarly, 2 typed when 22 is showing now produces 2, not another 22. So now all numbers act the same: if you want the number by itself, just type it, possibly more than once.

SOURCE CODE LIBRARIAN & REVISION CONTROL SYSTEM

TLIB™ keeps ALL versions of your program in ONE compact library file, even with hundreds of revisions!

- **Super Fast!** Updates (deltas) average **5-7 times faster** than PC/IX (Unix) SCCS. TLIB updates libraries faster than many editors load and save files!
 - **LAN-compatible!** Shared libraries with PC Network!
 - Synchronized control of multiple related source files.
 - Use with floppies or hard disk. TLIB doesn't need big temporary files, so you can maintain a 300K library on one 360K diskette, with room to spare, even with TLIB itself on the same disk. And libraries are more compact than with most other revision management systems.
 - Perfect for backup. Date and comments kept with each version. High data integrity because library data, once written, is never modified. Libraries are only appended, to minimize the chance of data loss due to a power glitch or hardware crash. And TLIB isn't copy-protected, either.
 - Free copy of Landon Dyer's excellent public domain **MAKE** utility. With macros, full source code. For DOS & VAX/VMS.
- PC/MS-DOS 2.x & 3.x **Just \$99.95 + \$3 s/h** Visa/MC

BURTON SYSTEMS SOFTWARE
P. O. Box 4156, Cary, NC 27511-4156
(919) 469-3068

Circle no. 212 on reader service card.

**DISK FORMAT
CONVERSION**

PC-DOS program
lets your PC
Read/Write/Format
over 275 formats

XENOCOPY-PC™
by Fred Calk

\$79.95 + \$5.00 S/H Sales Tax if CA.

Upgrades available from previous versions

To Order Contact:

XENOSOFT™

1454 Sixth Street, Berkeley, CA 94710

(415) 525-3113

Circle no. 225 on reader service card.

Analysis can serve as a touchstone to verify the accuracy of a solution reached by other means, and sometimes analysis can itself lead you toward the right factoring. But you can augment analysis with other approaches. Activate them deliberately by immersing yourself in the problem early on so that your translogical processes have time to play with the problem and deliver their results.

Forth's interactivity naturally leads to an experimental and exploratory approach that encourages an early intimacy with the characteristics and implications of a problem, which ultimately leads to a deep understanding. The feedback loop thus established often leads to long exchanges in which an idea is tried and gives a result that points to another idea: experience gives an insight on which to base a new attempt, producing a repeating cycle that moves to the heart of the problem.

Charles Moore, Forth's father, found the FORTRAN compiler he was using uncomfortable and awkward to use. He had the insight to see that, for his needs, it was factored incorrectly. The factoring of the FORTRAN compiler placed it outside the language. Moore saw that the correct factoring for his purposes put the compiler inside the language, where he could use it directly.

With the compiler now at hand, he factored it into its separate functions. He eliminated the complexities of parenthesis parsing by eliminating the parentheses. He made some words "immediate," to execute during compilation and thus function as compiler directives. (The directive *[COMPILE]*, for example, is immediate; it forces the following immediate word to be compiled even when it normally would execute.) He eliminated rarely used compiler constructs. The programmer could easily add them when they were needed, now that the compiler was a part of the language. Thus, in place of the old do-everything, batch-oriented compiler that stood outside the program, Moore built into the language a compiler factored into tools that the programmer could use in tailoring it to the current job.

CREATE and *DOES>* were found in the factoring of the compiler. These words give the programmer a strong

voice in compiler activities. The programmer's *CREATE... DOES>* words are added to the compiler and define new kinds of words targeted at the task at hand. These words owe their existence to the idea of factoring the compiler into the language, making it accessible for this kind of control.

CREATE and *DOES>* also factor the definition into phases. When the defining word is compiled (called its compile time), the (nonimmediate) words in its definition are laid down in the dictionary for later execution. Its run time comes when it is executed to define a child word; this is the child's compile time. At that time, the defining word's *CREATE* clause is executed, putting the child's definition into the dictionary. Only at the child's run time, when the child itself is executed, does the *DOES>* clause in the parent quicken at last to life.

Moore's factoring of the compiler also offers other words that can serve for defining words. The word *OCTAVE*, for example, doubly defined in Listing Three, page 94, might be used in a music application to define the frequency of a note an octave above a given note. The first definition of *OCTAVE* uses *CREATE* and *DOES>* as you would expect. *CREATE* puts a header into the dictionary, with *CREATE*'s usual code field. When the *CREATED* word is executed, its code field contributed by *CREATE* puts the address of the beginning of its parameter field on the stack.

The next step after *CREATE* (taken at the defined word's compile time) is to comma the number on the stack (at compile time) into the current top of the dictionary—the beginning of the parameter field of the new word. Comma advances the dictionary pointer past this parameter field. (Normally a *CREATED* word has no parameter field; here comma's action reserves the parameter field.)

The *DOES>* stored in *OCTAVE*'s definition terminates the compile-time action of the word being defined (when *OCTAVE* is executed to define a word). *DOES>* also replaces the *CREATE* code field in the child with a code field that points to itself.

When the child is run, its code field points to the *DOES>* in *OCTAVE*, and so it follows the dictates of its parent. *DOES>* now places on the stack the address of the child's parameter

Light a fire under your database.



We've been carrying a torch for dBASE for years.

So we're pleased to announce Quicksilver™, our totally new dBASE III Plus compiler.

Quicksilver produces native assembly code which runs dBASE applications like wildfire, faster than anything else on the market.

It also supports all the programming commands of dBASE III Plus, including multi-user syntax and memo fields. It includes a faster SORT and INDEX. And it uses the same data, index and

other files as dBASE.

For greater flexibility, Quicksilver lets you create your own User-Defined Functions. It also supports "dBASE Tools For C," and lets you link in your own C code for complete control of your database applications. (Our own powerful HOT C™ compiler is available for just \$99.)

With Quicksilver you can even create your own windowing environment, because we've added a full set of dBASE-like commands that give you up to 99 crisp, quick

windows without any other software.

And Quicksilver speed, power and compatibility are all yours for just \$599.

To order, call the WordTech Systems, Inc. HOTline at (415) 254-0900. But do it today. Because we all know that the race goes to the quick.

QUICKSILVER™
by WORDTECH

SOFTSTRIP® NOW OFFERS YOU SOMETHING IN NEVER BEFORE AVAILABLE...



CONVENTIONAL DATA HANDLING



THE SOFTSTRIP SYSTEM

A CHOICE.

Until now you were stuck with disks.

No more. Install our unique STRIPPER™ software on your personal computer today and discover the many benefits of the fastest, easiest, least expensive way to handle information.

STRIPPER lets you print — ON PAPER — your own machine readable Softstrip data strips using your dot matrix printer. The Softstrip System Reader reads that information into a computer rapidly. With STRIPPER and the reader, your PC and printer become part of the most versatile information handling system available.

With this system you can do anything you wish with any data you have in your PC — ON PAPER.

DATA ENTRY: Why use keystrokes when you can eliminate them with data strips? Whatever the document - invoices, packing slips, memos, letters, sales reports, the list is endless — simply print a data strip right on the same printed page. Now you have a document that is both human readable and machine readable. A typical document can be entered in only 15 seconds using data strips. And, it ends keystroke errors forever.

DATA DISTRIBUTION: Why copy disks? It's time consuming and expensive. Softstrip data strips will end all that. Simply photocopy as many data strips as you like and send them by mail. Data strips ignore folding, coffee stains, ink marks and, by the way,

magnetic fields. And if you're using telecommunications, you can stop making the phone company rich.

DATA STORAGE AND RETRIEVAL: Why have a file of disks and a file of paper? Eliminate one with Softstrip data strips. File the data strip with the document. Better still, print the strip right on the document. Then put it in a file or binder.

Retrieval is simple. To find existing data, pull the document and its related data strip from the file. They've been stored together. Then use the reader to enter the data. No more hassle trying to match documents with the right disk — if you can find it.

DATA TRANSFER: Why bother with cables, modems and phone lines to move files between computers? A Softstrip data strip generated by an IBM PC can be read into another PC, or compatible, an Apple or even a Macintosh. If you work at home on a Macintosh, make a data strip on your printer, take it into the office and read it into your IBM PC. Simple. And we've created the utilities to let you do that easily. (See Application Notes on opposite page.)

Fascinating, isn't it? Anything you can do with disks can be done with the Softstrip data strip system — faster, easier and at lower cost — ON PAPER.

All you need is STRIPPER software at **\$19.95** and the Softstrip System Reader at **\$199.95**.

DATA HANDLING

OCTOBER CASE HISTORY



A physical rehabilitation and pain management clinic in Hot Springs, Ark., has begun a trial program using the SOFTSTRIP System to reduce the huge volumes of medical paperwork in physician's offices, clinics and hospitals. Dr. Henrik Madsen, medical director and administrator of the Hot Springs clinic, is deeply involved in the management aspects of health care as well as medicine itself. One of the biggest problems facing the proper administration of medical offices, clinics and hospitals is paper. Voluminous patient records accumulate

rapidly, soon cramming all available shelf space. All of those records are hand or typewritten papers.

Using the Softstrip System, Dr. Madsen's clinic is recording patient records on data strips. Those strips are filed in place of the written medical records, significantly reducing the amount of shelf space needed, as well as handling times. In addition, recall of medical records to the clinic's IBM PC computer can be done quickly using data strips.

As Dr. Madsen observed: "Cost reduction is a vital national concern. If it can be accomplished in the administrative area, that many more dollars will flow to direct patient care. Everyone is looking for effective means to reduced administrative costs. This is one with promise."

APPLICATION NOTES

On the other side of this ad we said you can move data between different programs — on paper.

Using Softstrip data strips you can!

For instance move data between AppleWorks and Excel and back. Or Lotus 1-2-3 to and from AppleWorks.

We've created a series of several dozen Application Notes on Softstrip data strips. These lead you through simple steps to make the file transfer as easy as possible.

Here are just a few examples:

- AppleWriter to or from MacWrite.
- dBASE to or from Appleworks.
- WordStar to or from AppleWriter.
- WordStar to or from MacWrite.
- Framework to PageMaker.

And more are coming.

ACT NOW!! Don't delay.
See your local SOFTSTRIP
dealer or call us at 1-800-533-7323.
In Connecticut: 203-573-0150.

Users' Groups: Call for
special User Group deals.

These strips contain "The IRA Calculator," a worksheet for calculating returns with either Lotus or Excel. To receive the complete Application Note, call 1-800-533-7323, or write to Cauzin.

CAUZIN

835 South Main Street
Waterbury, CT 06706
(203) 573-0150


Softstrip

field. Then *DOES>* executes the phrase in the parent following itself until it reaches the parent's semicolon. In this example, the only word following *DOES>* is @, which replaces the address on the stack with the contents of that address: the doubled frequency stored when the child was defined.

The second definition of *OCTAVE* uses *CONSTANT* as the defining word. *CONSTANT* itself does everything we

need except double the number, and so we can eliminate *CREATE* and *DOES>* altogether.

Execution Arrays

A well-factored word can be used in unexpected ways because it is not enmeshed in the particularities of its original implementation. In the previous example, *CONSTANT* was used as a substitute for *CREATE* and *DOES>*. The next example uses the colon in an unexpected way by exploiting *DOES>*'s powers.

In Forth, an execution vector or ex-

ecution array contains the compilation addresses of Forth words. The program dips into the array using an offset somehow derived, fetches the address found there, and executes it. Different offsets can thus produce arbitrarily different results.

Execution arrays are a common tool when the users of a program select options from a menu. The natural implementation of a menu returns the number of the selected item, and this number can be used as the offset into an array of actions. The obvious approach is shown in Listing Four, page 94. *CREATE* puts the header *OPTIONS* into the dictionary. When *OPTIONS* is later executed, it puts on the stack the address of what amounts to the parameter field, the first byte after the header. In this example, several compilation addresses have been stored in the dictionary, beginning at this location. I started the compiler, and so the words following it were not executed. Instead, the compiler found their compilation addresses and stored them in the dictionary, word by word. The I turned the compiler off again. The words *>PRINTER*, *>DISK*, *>SCREEN*, and *>DOS* are assumed to have been defined earlier to perform the desired actions. The word *DO-OPTIONS* uses the number on the stack to dip into the array and execute the word thus referenced.

However, the action of first creating a header and then finding the compilation addresses of a series of words and storing those addresses into the dictionary as they are found is precisely the action of *:* (colon). I use *:* in a defining word that creates execution arrays, as shown in Listing Five, page 94.

In a typical defining word, *DOES>* terminates the actions that follow *CREATE* when the defining word's child is being compiled. *VECTOR*, however, contains no *CREATE*. The compilation begun by the colon continues until a semicolon turns off the compiler. As soon as the semicolon acts, the return stack takes the action back to the word being executed and continues with the next word in its definition. The word being executed (at compile time) is *VECTOR*; and the next word in its definition is *DOES>*. *DOES>* replaces the compilation address of the child, which this time

Do You Ever Get the Feeling That No One Speaks Your Language?

Arity/Prolog. The Language That Spans the Generation Gap.



Arity listens to what you ask for. You want a serious, versatile language that will go the distance for you. There are two very good reasons for you to use Prolog—to do your work smarter and faster. That's exactly what the Arity/Prolog development environment will help you do. Our powerful tools, based on the general purpose programming language Prolog, will significantly reduce your development time and allow you to solve a wide range of application problems.

No translation required Our development environment for the IBM PC family and all MS-DOS compatibles includes the Arity/Prolog Compiler and Interpreter, the Arity/Expert System, and Arity/SQL. And you can tie them all together. You can interface with several other programming languages and build extensions to your existing applications. You'll be truly multilingual—what better way to span the generation gap?

It can take you to new places You'll discover amazing speed, power, and flexibility using the Arity/Prolog programming environment, with its one gigabyte of virtual memory and fast, compact compiled code, for conventional applications. And if you're working in new territories, like expert systems or sophisticated database management systems, you'll be speaking the native tongue.

Speak it freely Our products are not copy protected and we charge no royalties, so you can use them in as many end-user applications as you'd like. Why keep the language of solutions all to yourself?

Join the thousands of assembly and C programmers who already use Arity/Prolog—the language of solutions.

Call 1-800-PC-ARITY Today.

Massachusetts residents call 617-371-1243.

ARITY CORPORATION

30 Domino Drive, Concord, MA 01742 U.S.A.

1-800-722-7489 or in Massachusetts call 617-371-1243

Software that roars.

FORTH LEADING THE NEW GENERATION OF COMPUTER LANGUAGES

The Forth Interest Group (FIG) is the association of programmers, managers and engineers who create practical, Forth-based solutions to real-world problems. FIG provides a climate of intellectual exchange to help each achieve his professional aims. Their uses of Forth include:

- Scientific Instrument Control
- Data Acquisition
- Metacompilation
- Stand-alone "Smart" Devices
- Expert Systems
- Business Accounting
- Graphics Systems



Each FIG member receives the bimonthly magazine *Forth Dimensions*. Its tutorials, innovative techniques, and extensive program listings are published exclusively for FIG members, and are refereed by premier experts in the Forth community.

The 8th Annual Forth Convention will be held at the new Doubletree Hotel in Santa Clara, California on November 21 and 22, 1986. Each year about 1,000 Forth enthusiasts attend the program of:

- Panel discussions by professionals who use Forth in their work
- Lectures by the field's most respected experts
- Exhibits by Forth Vendors

To join FIG or for further information, call us at **(408) 277-0668** or write P.O. Box 8231, San Jose, CA 95155, USA.

Circle no. 216 on reader service card.



Introducing The Most Important Programming Development Since The Introduction Of C...

**ADVANTAGE C++ For MS/PC-DOS,
Exclusively From LIFEBOAT.**

They say you can't be all things to all people. But Lifeboat's Advantage C++ proves that you can be! This exciting new product, developed by AT&T, represents a major programming breakthrough. By introducing the concept of classes, it enables programmers to use object-oriented programming methods. Plus it gives you a host of other major improvements over C.

Advantage C++ gives you the ability to create new functionality to solve your applications problems. It allows programmers to more productively build large and sophisticated applications. All the benefits of C, without its limitations.

Advantage C++ is available for the most popular C compilers, Lattice C and Microsoft C.

Why be limited to just C...When you can have all these pluses!

For a complete technical specification sheet call
1-800-847-7078
In NY: **914-332-1875.**

LIFEBOAT

The Full-Service Source for Programming Software.

Circle no. 118 on reader service card.

MULTI-TASKING!

UNO, DOS... MULTI-DOS!



**You can do this too,
if you have Multi-DOS.**

MultiDOS is the NEW Multi-Tasking Software that lets you run multiple programs on your PC all at the same time!

With Multi-DOS you can load all of your favorite programs (up to 32, limited by the size of individual programs and available memory) and switch from one program to another at a keystroke!

Compatible with most DOS software, including LOTUS, DBase, Wordstar, and others.

Multi-DOS \$19.95 +\$2.95 S/H

Not Copy Protected

for
Software Professionals

there's Multi-DOS Plus. DEVELOP
YOUR OWN MULTI-TASKING APPLI-
CATIONS!

- inter task message communication
- suspend task for specified interval
- execute external and internal tasks
- lock/unlock semaphores
- change task priority (8 levels)
- commands for suspend, resume, abort, etc.
- AND MORE!

Multi-DOS Plus \$29.95 +\$2.95 S/H

Not Copy Protected

Specifications and Requirements:

- IBM-PC/XT (or clone) with DOS 2.0 or later operating system.
- Multi-DOS occupies 42 kb of memory (48 kb for Multi-DOS Plus) and 4 to 16 kb of memory per active task.

ORDER NOW, call toll-free!

1-800-367-6707

VISA AND MASTERCARD ACCEPTED

Or send check or money order to:
**Nanosoft, 13 Westfield Rd
Natick MA 01760**

For Information or MA orders Call
(617)651-0091

MA orders add 5% sales tax. Outside
U.S.A. add \$7.95 S/H.

contains not the *CREATE* code field but the colon code field—the run-time colon that will normally execute the words in the definition in turn. *DOES*>, however, takes no notice of the contents of the code field. It simply overwrites them with the address of *VECTOR*'s *DOES*>.

When the child word *OPTION* is executed, *VECTOR*'s *DOES*> places the address of *OPTION*'s parameter field on the stack and swaps that address to find the number beneath. This number is doubled and then added to the parameter field address to get the compilation address of the word to be executed. That address is then fetched and executed.

VECTOR defines an execution array and also makes it produce and execute the right element of itself. Because the word itself does the work, the surrounding code is simplified. Harry Wilker, author of *Back to Basics Accounting*, which was written in Forth and is published by Peachtree, says that he begins a new program by figuring out what data structures he will need and what he wants these structures to do. He creates the appropriate defining words and writes his program outward from there, with the data structures themselves doing much of the program's work.

Wilker's approach is strong because it exploits some of Forth's special strengths and because it is theoretically correct: the program should indeed be rooted in the data structures. Probably more programmers would follow his example except, as noted earlier, many come to Forth from other languages and are accustomed to programming techniques that depend on developing algorithms rather than on defining new structures. We all are reluctant to discard a tool that once has worked.

Naming Style

VECTOR is, I think, a bad name: mechanical, klunky, and earthbound. In my July column I gave the name *FOR* to an array-defining word. As a name, *FOR* has what *VECTOR* lacks—*FOR* is a short English word that makes the code read naturally, and it doesn't belabor the reader with details of the implementation. For this

example, I think the name *EMPOWER* is vastly superior to *VECTOR*:—*EMPOWER* fits the task and has more panache. *VECTOR* was the working name; once the word is complete, it pays to take a minute to find a better name. I dub the word *EMPOWER*.

Naming, however, takes on some aspects of style, and matters of taste seldom find unanimity. Many Forth programmers find names such as *FOR* and *EMPOWER* about as agreeable as eggshell in a soufflé. They prefer names such as *ARRAY* and *VECTOR*, which they find direct and descriptive; *FOR* and *EMPOWER* strike them as pretentious and ethereal, abstract and unrelated to what is happening. I suspect that they also find names of this ilk to be inappropriately playful in a programming context.

I, of course, believe that I tread a middle course of elegance and economy. For example, I rejected the idea of defining *with* as an immediate no-op to be used with *EMPOWER*:

EMPOWER *OPTION* with >PRINTER...

(*with* must be immediate so that it will leave no tracks in the compiled definition). I also take comfort from a name such as *DROP*, which also speaks to the idea of what is happening rather than to the mechanism that does it. *DROP* grasps the metaphoric center of the action.

But it is only fair to recognize that some Forth programmers shudder when they encounter names that delight others by the unexpected aptness of a word found in a new but fitting context. Some find delight in wordplay; others do not. For both groups, though, names are indeed important—on that they agree. In *Thinking Forth* (Englewood Cliffs, N.J.: Prentice-Hall, 1984) Leo Brodie offers sound advice on choosing names: choose names according to "what" not "how"; find the most expressive word; favor short words; hyphenated names may be a sign of bad factoring; and many others, with plenty of examples.

Correction vs. Prevention

Can you tell what the following sequence means and give the next two numbers? (No peeking.)

0 0 9210 1023 -1 -30721

C & PASCAL PROGRAMMERS

Blaise Computing provides a broad range of programming tools for Pascal and C programmers, with libraries designed for serious software development. You get carefully crafted code that can be easily modified to grow with your changing needs. Our packages are shipped complete with comprehensive manuals, sample programs and source code.

C TOOLS PLUS

\$175.00

NEW! Full spectrum of general-purpose utility functions; windows that can be stacked, removed, and accept user input; interrupt service routines for resident applications; screen handling including EGA 43-line text mode support and direct screen access; string functions; and DOS file handling.

PASCAL TOOLS/TOOLS 2

\$175.00

Expanded string and screen handling; graphics routines; easy creation of program interfaces; memory management; general program control; and DOS file support.

VIEW MANAGER

\$275.00

Complete screen management; paint data entry screens; screens can be managed by your application program; block mode data entry or field-by-field control. Specify C or IBM/MS-Pascal.

ASYNCH MANAGER

\$175.00

Full featured asynchronous communications library providing interrupt driven support for the COM ports; I/O buffers up to 64K; XON/XOFF protocol; baud rates up to 9600; modem control and XMODEM file transfer. Specify C or IBM/MS-Pascal.

Turbo POWER TOOLS PLUS

\$99.95

NEW! Expanded string support; extended screen and window management including EGA support; pop-up menus; memory management; execute any program from within Turbo Pascal; interrupt service routine support allowing you to write memory resident programs; schedulable intervention code.

Turbo ASYNCH PLUS

\$99.95

Complete asynchronous communications library providing interrupt driven support for the COM ports; I/O buffers up to 64K; XON/XOFF protocol; and baud rates up to 9600.

RUNOFF

\$49.95

NEW! Text formatter written especially for programmers; flexible printer control; user-defined variables; index generation; and general macro facility. Crafted in Turbo Pascal.

EXEC

\$95.00

Program chaining executive. Chain one program from another even if the programs are in different languages. Shared data areas can be specified.

ORDER TOLL-FREE 800-227-8087!



BLAISE COMPUTING INC.

2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

Circle no. 217 on reader service card.

Dr. Dobbs' Journal, October 1986



NEW! FROM BLAISE COMPUTING

Today's programmers need more than yesterday's tools. Requirements such as removable windows and "sidekickable" pop-up utilities are changing the face of program design. You need to filter interrupts so that other resident programs still work. You need the ability to switch between multiple display pages and monitors. Today's technical demands are almost endless, but C TOOLS PLUS gives you what you need.



SOLID LIBRARY SUPPORT

Blaise Computing offers you solid library support that can meet all your demands and more. C TOOLS PLUS embodies the full spectrum of general-purpose utility functions that are critical to today's applications.

Here's just part of the PLUS in C TOOLS PLUS:

- ◆ **C TOOLS and C TOOLS 2** compatibility—two packages that receive rave reviews for quality, organization, usability and documentation.
- ◆ **FULL SOURCE CODE**

C Tools PlusTM

For The Programmer Whose Alphabet Begins & Ends With "C"



- ◆ **WINDOWS** that are stackable, removable, that support word wrap and that can accept user input.
- ◆ **INTERRUPT SERVICE ROUTINE** support for truly flexible, robust and polite resident applications.
- ◆ **MULTIPLE** monitor and display support, including EGA 43-line mode.
- ◆ **FAST DIRECT VIDEO ACCESS** for efficiency that will not constrain good program design.
- ◆ **DOCUMENTATION, TECHNICAL SUPPORT** and attention to detail that have distinguished Blaise Computing products over the years.

C TOOLS PLUS supports the Microsoft (and IBM) 3.00 and Lattice 3.00 C compilers and is just \$175.00.

Also Available Are:
C VIEW MANAGER—A kit for building data entry screens and menus. Begin by designing on-screen what the operator will see; call upon our library functions from your program to display the screens and retrieve the data. Just \$275, including all library source code.

C ASYNCH MANAGER—provides the crucial core of hardware interrupt support needed to build applications that communicate. It

also includes the "XMODEM" file-transfer protocol and support for Hayes-compatible modems. All source code is included for \$175. **C TOOLS & C TOOLS 2**—an indispensable combination still available at a low price of \$175, including all source code. See review in PC Tech Journal, 6/85.

BLAISE COMPUTING INC.

2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

ORDER TOLL-FREE 800-227-8087!

YES, send me the PLUS I need! Enclosed is \$_____ for C TOOLS PLUS. (CA residents add 6½% Sales Tax. All domestic orders add \$10.00 for Federal Express shipping.)
Name: _____ Phone: (____) _____
Shipping Address: _____ State: _____ Zip: _____
City: _____ Exp. Date: _____
VISA or MC #: _____

STRUCTURED PROGRAMMING

(continued from page 120)

These numbers turn out to be useful in an application I wrote recently. Following is a description of how they arose. (The next two numbers are, of course, 1 and 10240. Let me know if this is too easy.)

I mentioned earlier how Moore simplified the handling of parentheses in arithmetical expressions by eliminating parentheses. The same technique works well in other con-

texts. In a program I am working on now, for example, the user is asked to enter a file name. In PC-DOS and MS-DOS, all characters in the ASCII character set are equal but some are more equal than others. The less-equal ones are not allowed in file names. Some of the less-equal characters can be used, but they limit DOS activities—for example, an embedded blank in a file name prevents *COPY* and *DEL* from performing their functions.

The first approach that occurred to

me was that I should edit the file name for a new file and warn the user when an illegal character had been used, asking for correction or reentry. Then I realized this was a poor factoring of effort. Why not write code to keep the illegal characters from being entered in the first place, rather than to detect and fix them later?

I set up a bit array 16 bytes long (128 bits) and turn on the bits corresponding to the ASCII values of the legal file-name characters. When the user enters the file name, I simply don't accept any character for which the bit is off. A little checking saved me from having to write a (more complex) routine that would detect errors after the fact and also saved me from having to figure out a good interface to communicate errors to the user and collect corrections (or allow the user to quit).

My first solution was to set the array bits in an initialization word, which used the bit words shown in Listing Six, page 94. (These are reprinted from my last column, with one name improvement.) But then I realized that setting the bits took more room than the bit table itself, so I removed the bit setting from the program (Listing Seven, page 94). Note that I don't use the lowercase alphabet: When the user is entering file names, the program shifts any lowercase letters to uppercase.

After using *READOUT* to verify the correctness of the bits, I used *READ* to list the equivalent sequence of numbers—the sequence of eight numbers shown at the beginning of this section. These are used to create an array, as shown in Listing Eight, page 94. *Voilà*: no need to edit the file name because illegal information is barred at the door.

The bit words in Listing Six were factored differently when I first wrote them. *AIM* was not originally included in the bit words (*+BIT*, *-BIT*, and so on). It was only after I used the words for a while that I realized that *AIM* should be factored into the bit operators. By putting *AIM* inside the words, I hide that particular operation.

I have discussed how Forth represents a new way to factor a compiled language, with the compiler factored into the language. I have also talked about how the compiler itself is fac-

Epsilon by Lugaru Software, Ltd.

The most advanced, customizable programmer's editor you can buy.

True Concurrency. Don't be fooled by misleading claims of concurrency from other vendors: this is *true concurrency*, not some simple "run the program and return" facility. Run compilers, linkers, etc., inside Epsilon and they run *as you work*. Epsilon's concurrency integrates buffer management and program I/O. The full power of Epsilon is always available to edit program input and output. You're free to edit other things while these programs run.

Ultimate Customizability. You may have seen some other vendors hinting that they have a C-like extension language. Don't be fooled. Ask them if their extension language has the syntax and types of C. Epsilon uses an embedded C interpreter for its extension language, complete with all the data types and operators of C. This is a *real language*, not an afterthought. All of Epsilon's commands were written in it, and it's fast!

No-Nonsense Help. Some editors offer a help screen or two listing the most basic commands—useless after the first week. Others spit out everything from a fixed file—which means they can't tell you about your changes to the keyboard, or commands you've added. We think an editor that claims to be reconfigurable should be smart enough to reflect your changes in its help system. Epsilon's extensive help system can tell you what commands apply to files, what a certain command does, which keys you can use to invoke it, what a certain key does. . . . Our help system remains helpful even if you make changes.

- | | |
|--------------------------------|--------------------------------|
| • Fast | • EMACS style command set |
| • Concurrent Processes | • Context Sensitive Help |
| • Multiple Windows | • Regular Expression Search |
| • Unlimited File Size | • Unlimited Number of Files |
| • On-line Tutorial | • File/Command Name Completion |
| • Automatic Swap File | • Convenient Keyboard Macros |
| • Supports Large Displays | • Directory Perusal |
| • Saves Deleted Text (n times) | • Uses All Available Memory |

Our Guarantee to You. We know it's difficult to find software that works for you. That's why we made Epsilon the most customizable editor on the market. And that's why we offer a *30 day return privilege* with a complete refund. We're confident that once you've tried Epsilon, you won't go back.

So call (412) 421-5911 to order Epsilon at no risk using your Visa, MasterCard, or American Express card. Company PO's and C.O.D. orders are also welcome. Epsilon runs on 256K IBM PC/XT/AT's, and costs only \$195.00.

Lugaru

Lugaru Software Ltd.
5740 Darlington Road
Pittsburgh, PA 15217
(412) 421-5911

Circle no. 135 on reader service card.

LOGITECH MODULA-2/86 HOLIDAY PACKAGE

\$89 Price

- Separate Compilation
- Native Code Generation
- Large Memory Model Support
- Multitasking
- Powerful Debugging Tools
- Comprehensive Module Library
- Available for the PC and the VAX

Use LOGITECH MODULA-2/86 to decrease your overall development cycle and produce more reliable, more maintainable code.

LOGITECH MODULA-2/86 **\$89**

Includes Editor, Run Time System, Linker, 8087 Software Emulation, Binary Coded Decimal (BCD) Module, Logitech's comprehensive library, Utility to generate standard .EXE files. AND more!

LOGITECH MODULA-2/86 with 8087 Support **\$129**

LOGITECH MODULA-2/86 PLUS **\$189**

For machines with 512K of RAM. Increases compilation speed by 50%.

RUN TIME DEBUGGER (Source level!) **\$69**

The ultimate professional's tool! Display source, data, call chain and raw memory. Set break points, variables, pinpoint bugs in your source!

UTILITIES PACKAGE **\$49**

Features a Post-Mortem Debugger (PMD). If your program crashes at run-time the PMD allows you to analyze the status of the program and locate the error. Also includes a Disassembler, Cross Reference Utility, and Version that allows conditional compilation.

LIBRARY SOURCES **\$99**

Source code now available for customization or exemplification.



WINDOW PACKAGE **\$49**

Build windows into your programs. Features virtual screens, color support, overlapping windows and a variety of borders.

MAKE UTILITY **\$29**

Figures out dependencies and automatically selects modules affected by code changes to minimize recompilation and relinking.

CROSS RUN TIME Debugger and ROM Package **\$199**

Still available at an introductory price!

TURBO PASCAL to MODULA-2 TRANSLATOR **\$49**

"Turbo Pascal... is a very good system. But don't make the mistake of trying to use it for large programs."

*Niklaus Wirth**

Our Translator makes it even easier for Turbo users to step up to Modula-2/86. It changes your Turbo source code into Modula-2/86 source, solves all the incompatibilities, and translates the function calls of Turbo into Modula-2/86 procedures. Implements the complete Turbo libraries!

Call for information about our VAX/VMS version, Site License, University Discounts, Dealer & Distributor pricing.

30 Day Money Back Guarantee!

To place an order call our special toll free number:

800-231-7717
in California
800-552-8885

\$199

**Special
Holiday Offer**

Step up to the power of LOGITECH MODULA-2/86 at a saving of nearly \$100 off our usual low prices! We're offering a complete tool set including our compiler with 8087 support (for use with or without an 8087), our Turbo to Modula-2/86 Translator, Run Time Debugger, and Utilities in one holiday package at a special price!

YES I want to step up to LOGITECH MODULA-2/86!

Here's the configuration I'd like:

- | | |
|---|--------------|
| <input type="checkbox"/> Special Holiday Package | \$199 |
| <input type="checkbox"/> Logitech Modula-2/86 | \$89 |
| <input type="checkbox"/> with 8087 support | \$129 |
| <input type="checkbox"/> Plus Package | \$189 |
| <input type="checkbox"/> Turbo to Modula Translator | \$49 |
| <input type="checkbox"/> Run Time Debugger | \$69 |
| <input type="checkbox"/> Utilities Package | \$49 |
| <input type="checkbox"/> Library Sources | \$99 |
| <input type="checkbox"/> Window Package | \$49 |
| <input type="checkbox"/> Make Utility | \$29 |
| <input type="checkbox"/> ROM Package | \$199 |

Add \$6.50 for shipping and handling. Calif. residents add applicable sales tax. Prices valid in U.S. only.

Total Enclosed \$ _____

☐ Visa ☐ MasterCard ☐ Check Enclosed

Card Number _____ Expiration Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Phone _____



LOGITECH

Logitech, Inc.
805 Veterans Blvd.
Redwood City, CA 94063
Tel: 415-365-9852

In Europe:
Logitech SA, Switzerland
Tel: 41-21-879656
In Italy: Tel: 39-2-215-5622

STRUCTURED PROGRAMMING (continued from page 122)

tored into a variety of words that can be used in new combinations for new results. And now I factor "regular" Forth words to hide complexity where appropriate. Factoring is a part of Forth at every level.

How can you improve your factoring skills? The answer really is, "Try." I have outlined some methods in this article, but like any intuitive skill, factoring is developed through experience. There is no algorithm that will invariably arrive at the right

result. Like the right word in a poem, the right factoring comes with a click. You may approach it step by step, but ultimately you make a leap. Once there, it is easy to build a bridge of analysis back to where you were. Leaping is learned by trying to leap.

Factoring in fact is not merely a Forth issue nor even limited to programming. Two good books on the processes that factoring involves concern arts other than programming. These books are *Writing Without Teachers* by Peter Elbow (New York: Oxford University Press, 1973) and *The Art of Craft* by Carla Needleman (New York: Avon Books, 1981). Probably the best discussion of Forth factoring is in Brodie's book *Thinking Forth*, mentioned earlier. Richard Bolles wrote a book (*The Three Boxes of Life*, Berkeley, Calif.: Ten Speed Press, 1981) in which he discusses how people factor their lives. He believes that people incorrectly factor learning, work, and play (retirement, for example) into three separate times in their lives, and he explores other factorings to stretch the three as strands lengthwise along your life.

Good factoring is generally the result of a creative insight that transforms your view of the problem. Edward de Bono has written a variety of books that directly address *lateral thinking*, which is his term for this mode of thought. Some of his books I have liked are *The Mechanism of Mind* (1971), *Po* (1974), *The Five-Day Course in Thinking* (1974), and *The Use of Lateral Thinking* (1975). All are published by Penguin Books, Middlesex, England, but are readily available in this country as well. Pergamon Press (Fairview Park, Elmsford, NY 10523) publishes the CoRT Program, a six-lesson course in creativity and thinking, which was developed by de Bono and his associates. CoRT is an acronym from The Cognitive Research Trust.

Outline Processors and Programming

I have found that an outline processor is a great tool for analyzing a programming problem. Outline processors seem particularly suited to a language such as Forth, whose words form a hierarchy that can mimic the structure of the outline. The outline develops naturally from a top-down

analysis.

I use an outline processor to sketch the overall shape of a program, but it is also very useful ad hoc. When I get stuck, or find myself lost or confused, I call on my trusty outliner. I aim the outline just at the problem of the moment. A short session of outlining usually produces a "script" that I can take back to Forth and use to direct the code I write. The Forth, of course, is written from bottom up, or inside out—in the inverse order of the outline structure. In writing the outline, I don't attempt to write Forth commands. I simply write simple English statements that describe what needs to be done as clearly as I can.

My first pass through the problem usually produces a rambling set of statements in the wrong order. The advantage of the outline processor is the ease with which I can revise and reorder the topics, insert new topics, and move topics (with or without their subtopics) up or down in the hierarchy.

The outline processor I like most is MaxThink [MaxThink, 230 Crocker Ave., Piedmont, CA 94610; (800) 227-1590, in CA (800) 642-2406]. MaxThink costs \$89 and is available for the IBM PC and compatibles and for the Apple Macintosh. It is not copy protected. In addition to the usual set of outline commands and a good user interface, it has a variety of tools to stimulate thought. Neil Larson, the designer, is a longtime fan of de Bono, and some of de Bono's techniques are built into this package. Framework II is also a nice package, but because it is copy protected, I would not base any important work on it.

One nice side effect of outlining the solutions is that the outlines become valuable adjuncts to the program documentation. Also, the use of outlines allows you to tackle a larger problem than you could otherwise manage: the outline marks the trail and organizes the effort, letting you focus your concentration on the parts without losing a grasp of the whole.

Things Mom forgot to Mention

After writing a large program, we are painfully aware of oversights and missed opportunities. Our most pow-



Users'
Group

Over 90 volumes of public domain "C" software including:

- compilers
- editors
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details

The C Users' Group

Post Office Box 97
McPherson, KS 67460
(316) 241-1065

Circle no. 181 on reader service card.

ICs

PROMPT DELIVERY!!!
SAME DAY SHIPPING (USUALLY)
QUANTITY ONE PRICES SHOWN FOR AUGUST 24, 1986

OUTSIDE OKLAHOMA: NO SALES TAX

DYNAMIC RAM			
1Mbit	1000Kx1	100 ns	\$69.50
4464	64Kx4	150 ns	4.20
41256	256Kx1	100 ns	5.10
41256	256Kx1	120 ns	3.10
41256	256Kx1	150 ns	2.93
41128	128Kx1	150 ns	4.99
4164	64Kx1	150 ns	1.45
EPROM			
27512	64Kx8	250 ns	\$24.00
27C256	32Kx8	250 ns	7.00
27256	32Kx8	250 ns	5.46
27128	16Kx8	250 ns	4.15
27C64	8Kx8	200 ns	5.15
2764	8Kx8	250 ns	3.75
STATIC RAM			
43256L-12	32Kx8	120 ns	\$33.00
6264LP-15	8Kx8	150 ns	3.10

OPEN 6 1/2 DAYS 7AM-9PM. SHIP VIA FED-EX ON SAT.
SUNDAYS & HOLIDAYS: SHIPMENT OR DELIVERY VIA U.S. EXPRESS MAIL

SAT DELIVERY INCLUDED ON FED-EX ORDERS RECEIVED BY: Th: Std Air \$6/4 lbs Fr: P-One \$13/2 lbs

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts
MICROPROCESSORS UNLIMITED, INC.
24,000 S. Peoria Ave.,
BEGGS, OK. 74421 **(918) 267-4961**

Please call for current prices because prices are subject to change. Shipping & insurance extra. Cash discount prices shown. Orders received by 9 PM CST can usually be delivered to you the next morning, via Federal Express Standard Air (\$6.00) or Priority One (\$13.00).

640 Kbytes MOTHERBOARD KITS: Zenith 150 \$77.74
IBM PC/XT, Compat Portable & Plus: hp Vectra

V20 8Mhz \$12.00
V30 8Mhz \$14.00
8087 \$102.00
80287-8 8Mhz \$335.00
5 Mhz \$102.00
8Mhz \$102.00

Circle no. 105 on reader service card.

erful tool is hindsight, and the look backward can often provide good guidance for the next project. Experience is not one big thing; it is many little things. Here are a few words I wish I had read before starting my latest project.

Screen files are easier to use than text files because you can load and test individual screens. However, because you don't have to deal with the total file as you work, a screen file grows without your realizing it—especially if you save a precompiled version of the work to date so that you are always adding on just a little bit.

I am glad I used screen files, but I wish I had paid closer attention to factoring the functions into different files instead of making the files match the program modules. The date routine, for example, should have been in a file by itself, to be included whenever I needed it. Instead, it is copied into the different module files—bad show. Next time I will use many, many little files.

The gradual accretion that builds the file also puts more code behind you than you realize. Next time I will take a weekly break to revisit and spruce up all the code written during the week. Hindsight is so powerful, I will arrange to use it early and often. My solemn pledge is that during this weekly review I will do the following:

- rearrange the code so the screens are easy to read, adding new screens when more space is needed;
- ponder the names I have chosen and see if they can be improved;
- verify the accuracy of the stack comments;
- add comments to the code to explain not what it is doing but why;
- try to improve the way the words are factored and in particular to find factorings that produce useful tools.

Do you have any resolutions to add to this list? Send them in; maybe we can make a poster for Forth programmers.

DDJ

(Listings begins on page 94.)

Vote for your favorite feature/article.
Circle Reader Service No. 7.

UNIX Tools on DOS

MKS Toolkit

NOW WITH vi

Over 70 programs that perform tasks on machines like the IBM PC, XT, or AT with the ease that one would expect while working under UNIX. Designed especially for those developing software in a DOS environment, these utilities include:

- awk** — data transformation & report generation language
- prof** — give a profile of the execution times of a command
- egrep** — find a string using full regular expression patterns
- diff** — find the differences between two files

cat	chmod	cmp	comm	cp	cut	date	dd	dev
df	du	echo	ed	file	find	head	help	join
lc	line	ls	more	mv	nm	od	paste	pwd
rm	sed	sh	size	sort	split	strings	tail	time
touch	tr	uniq	wc					and more ...

The programs come with a shell and complete UNIX-style command-line file name expansion on 3 DSDD 5.25" floppies, load and run under DOS, and are not copy-protected. Phone support is available during business hours. Full documentation is included.

Price: \$139 from:

Mortice Kern Systems Inc.,

43 Bridgeport Rd. E., Waterloo, Ontario N2J 2J4

519-884-2251

For information or ordering call collect:

MasterCard & VISA orders accepted. OEM & dealer inquiries invited.

UNIX is a trademark of Bell Labs. MS-DOS is a trademark of Microsoft Corp.

STREAMLINE YOUR PROGRAMMING

Circle no. 249 on reader service card.



The Great **awk** is now in the MKS Toolkit!

Brand New From Peter Norton A PROGRAMMER'S EDITOR

only
\$50

that's *lightning fast* with the *hot* features programmers need

Direct from the man who gave you *The Norton Utilities*, *Inside the IBM PC*, and the *Peter Norton Programmer's Guide*.

THE NORTON EDITOR

"This is the programmer's editor that I wished I'd had when I wrote my *Norton Utilities*. You can *program your way to glory* with *The Norton Editor*."

Peter Norton



*Easily customized, and saved
Split-screen editing
A wonderful condensed/outline display
Great for assembler, Pascal and C*

Peter Norton Computing, Inc., 2210 Wilshire Boulevard,
Santa Monica, CA 90403, 213-453-2361. Visa,
Mastercard and phone orders welcome.

The Norton Editor™ is a trademark of Peter Norton Computing, Inc. © 1986 Peter Norton Computing

THE PROFESSIONAL PROGRAMMER

Professional Organizations

The most well-known professional organizations for programmers are the ACM (Association for Computing Machinery) and IEEE (Institute for Electrical and Electronics Engineers), each of which has many chapters, SIGs (special-interest groups), and publications. There are also many state, regional, and local groups that help software developers deal with laws on business practices, copyright, trade secrets, and so on. Many of these groups work actively to influence legislation and generally promote the cause of the software developer.

The Massachusetts Computer Software Council

(MCSC), which was founded in 1985, is a nonprofit industry association of chief executive officers of independent software companies in Massachusetts. The main goal of this group is to represent the interests and viewpoints of its members and their businesses. Membership is open to all CEOs of businesses principally engaged in the design, development, or distribution of computer software products or services and whose primary place of business is in Massachusetts. Founding members of MCSC include Mitchell Kapur, CEO of Lotus Development Corp., and David Bricklin, founder of VisiCalc. The group meets quarterly to discuss issues of importance to

Massachusetts' software industry. Dues are based on the size of the company. A membership newsletter, *Software Council News*, is published quarterly.

The Washington State Software Industry Development Board (WSSIDB) is sponsored by the Economic Development Partnership. WSSIDB sponsors seminars that focus on issues of interest to software companies, including topics on money, legal issues, and marketing. Some SIGs of WSSIDB include the Education Committee, the Consultants and Entrepreneurs Group, and the Northwest Venture Club. *Software Board News* is published quarterly for due-paying members.

The Software Entrepreneurs' Forum (SEF) meets in the Silicon Valley area. The monthly seminars cover such topics as legal, tax, and marketing issues, as well as future trends in the computer industry. SIGs sponsored by SEF include Macintosh, CD-ROM, Vertical Markets, IBM Technical, and Marketing. The SIGs meet monthly or bimonthly. A monthly newsletter is sent to all members.

The speakers at this summer's SEF meetings covered a variety of interesting topics. Paul Davis of Microsoft gave an overview of the Windows system and discussed the benefits of Windows to developers in terms of I/O device independence, compatibility with next generation CPUs, and overcoming the 640K barrier. Guy Kawasaki of Apple Computers gave listeners an outline of Apple from a developer's point of view. He discussed the key to success in the Apple marketplace and explained the

concept of software evangelism: getting people excited about doing Apple development. Andy Hertzfeld, who created much of the Macintosh system software, as well as ThunderScan and Switcher, discussed his new product Servant, which is designed to replace Finder.

For more information on these professional organizations, please contact them at the following addresses:

Association for Computing Machinery (ACM)
11 W. 42nd St.
New York, NY 10036
(212) 869-7440

The Institute of Electrical and Electronics Engineers (IEEE)
345 E. 47th St.
New York, NY 10017
(212) 705-7589

Massachusetts Computer Software Council (MCSC)
c/o MicroMentor Inc.
124 Mount Auburn St.
Cambridge, MA 02138
(617) 497-5716

Software Entrepreneurs' Forum (SEF)
P.O. Box 61031
Palo Alto, CA 94306
(415) 854-7219

Washington State Software Industry Development Board (WSSIDB)
c/o The Economic Development Partnership
18000 Pacific Highway S.
Seattle, WA 98188
(206) 433-1613

DDJ

Vote for your favorite feature/
article.
Circle Reader Service No. 8.

Z80, 8085, HD64180 "Cross" Assemblers!!

Our Z80 "cross" assembler for the IBM-PC and PC/AT blows the competition away. We run **15 times** faster than either of our competitors*. Yet, our assembler is much more powerful, with many extra features. Our secret is The Wizard who spent 3 years hand-coding what is probably the world's most efficient assembler and linker. We had to build a Z80 co-processor board to put The Wizard's code on your PC. He codes in Z80 assembly language, just like you. The plug-in co-processor board is included with each assembler-linker for only \$295. If you're using one of those slow guys, now is the time to upgrade. We'll give you 2 assemblers for \$495.00 to make it easier. Specify if you want the Z80/HD64180 version or the 8085 version.

Z-World	\$295.00
2500AD	\$199
Avocet	\$349
1300 Lines/Minute 20300	

Z-World
2065 Martin Ave. #110
Santa Clara, CA 95050
(408) 980-1678



* Request our test report.

Circle no. 162 on reader service card.

MIX C COMPILER



**Works on all
MSDOS/PCDOS
and all CP/M Z80
COMPUTERS!
(Not Copy
Protected!)**

The Powerful Mix
C COMPILER
*Harness the Power of the C Language
with this full featured Compiler*

Incredible Value

\$39⁹⁵
AT ONLY With 30
Day

Money-Back Guarantee

C FEVER *catch it!*

It's becoming an epidemic... everyone is switching to C! First there were a few hackers, then came the college students, next the major software houses, and now the rest of the programming world. Programmers everywhere are infected with the desire for SPEED, POWER, and PORTABILITY.

It's time to face the inevitable. You're going to catch the fever too! When you do, give us a call. We've got the best cure—an illustrated guide to the C language, plus a complete program development system. Everything you need to master the C programming language... all at a price that's less than the cost of a book!

But don't let this price fool you. Our system is powerful; it compiles twice as fast as the others, is completely standard, and it's very easy to use. Most C compilers are designed for wizards. We have designed ours for you!

What do you get for a mere \$39.95?

- A 312 page book filled with sample programs, plus
- A fast, standard, full featured C compiler that supports all data types and the latest features like bit fields, enumerations, structure assignment, and passing/returning structures.
- A file linker that loads separately compiled files, searches libraries, and builds an executable program.
- An extensive library of over 300 C functions (including the standard C functions and the computer specific functions that provide direct access to the operating system and BIOS).
- Tools that allow you to optimize your programs for minimal space or maximum speed.

Operators are standing by... Please use this Number for ORDERS ONLY!

CALL TOLL FREE FOR RUSH ORDER DELIVERY!

1-800-523-9520

**IN TEXAS, PLEASE
CALL TOLL FREE
1-800-622-4070**

For Technical Support Please call 1-214-783-6001

MIX Software, Inc. / 2116 E. Arapaho / Suite 363 / Richardson, Texas 75081

Or contact our Worldwide Distributors direct in:

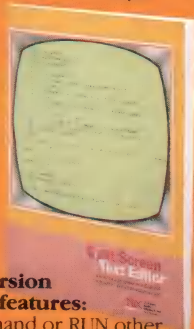
Canada: Saraguay Software 1-416-923-1500 Switzerland: DMB Communication CH-1-825-53-29
Australia: Techflow 047-586924 France: Info/Tech 1-43-44-06-48

Split Screen Text Editor

**an Incredible
Value \$29⁹⁵
AT ONLY**

(Not Copy Protected!)
Works on all MSDOS,
PCDOS and CP/M Z80
Computers

Our high powered editor is great for editing high level languages. **It works just like Micropro's Wordstar[®]** but macros allow you to create your own custom editor, and the split-screen feature lets you edit two files at the same time.

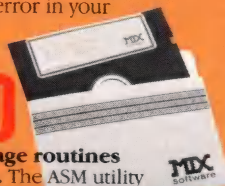


The MSDOS/PCDOS version is loaded with special features:

- Execute any DOS command or RUN other programs from the editor.
- Quickly edit files as large as 300,000 characters.
- Compile MIX C programs directly from memory. The editor automatically positions the cursor to the first error in your program.

ASM UTILITY
**an Incredible
Value AT ONLY \$10**

Call assembly language routines from your C programs. The ASM utility works with Microsoft's MASM or M80 assembler. Macros make it easy! Works just as if you were calling a C function, and you can even call C functions from assembly language. Lots of useful assembly language functions are included as examples.



RUSH REPLY ORDER FORM!

SPECIAL OFFER!

Buy both for an even greater value!

SAVE \$14.95 Off Our Regular List Price!

Limited Time Only
\$54⁹⁵
C Compiler & Text Editor

Please check method of payment:

☐ Check ☐ Money Order ☐ MasterCard/VISA

Your Card #: _____

Expires _____

Shipping Charges: (No charge for ASM Utility)

In the U.S.A.: Add \$5.00 per Order.

In CANADA: Add \$10.00 per Order.

OVERSEAS: Add \$10.00 per Text Editor. Add \$20.00 per C Compiler. Add \$30.00 for combined C Compiler and Text Editor.

Operating System: (Check one)

☐ CP/M Z80 ☐ MSDOS/PCDOS

Specify Your Computer Name _____

Specify Disk Format _____

NAME _____

Telephone A/C (_____) _____

Street _____

City _____

State _____

Country _____ ZIP _____

**MIX
software**

2116 East Arapaho
Suite 363
Richardson, Texas, 75081

Ask about our Volume Discounts!

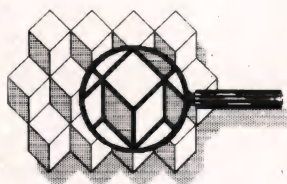
Call 1-214-783-6001

D

Circle no. 156 on reader service card.

Description	Quantity	PRICE	Total Order
Split-Screen Text Editor	_____	\$29.95	\$_____
C Compiler	_____	\$39.95	\$_____
C and Text Editor (Special)	_____	\$54.95	\$_____
ASM Utility	_____	\$10.00	\$_____
Texas Residents Add 6.125% Sales TAX		\$_____	\$_____
Shipping Charges (See at Right)		\$_____	\$_____
TOTAL OF YOUR ORDER:		\$_____	\$_____

OF INTEREST



The mass-storage industry finally seems to be perking up again, now that CD-ROMs (and other optical data-storage devices) are becoming commercially feasible. Prices of high-capacity, high-speed disk drives are dropping again, and new advances are being made in recording technology. The smallest hard disks of all—the ones that live on their own interface cards—are starting to show some pretty impressive storage figures, and the data densities of the physically larger devices are increasing just as rapidly. Here are some examples:

Express Systems offers a range of hard disks on cards for the IBM PC and PC/AT. They're called Express Hard DiskCards, and capacities range from 20 megabytes to 60 megabytes.

Prices range from \$449 to \$1,095. Reader Service No. 16.

Express Systems
1254 Remington Rd.
Schaumburg, IL 60195
(312) 882-7733 x3600

A 21-megabyte hard disk on a card, called SlotMachine, is available from **Kammerman Labs**. It fits into a standard IBM PC slot and sells for \$499. Reader Service No. 17.

Kammerman Labs Inc.
7861 S.W. Cirrus Dr.
Beaverton, OR 97005
(800) 522-2237

Instar Corp. sells an optical disk software package called ODI-PC that lets an IBM PC randomly access up to 1 gigabyte of storage on a single removable disk. The package requires a write-once, read-many (WORM) optical disk drive from Alcatel-Thompson, Xerox, or Sony. The interface card, cable, manual, and software list for \$1,575. Reader Service No. 18.

Instar Corp.
141-6815 8th St. NE
Calgary, AB
Canada T2E 7H7
(403) 275-3143

The DK815-10 is an 8-inch hard-disk drive from **Hitachi** that stores 1,050 megabytes of unformatted data. The unit uses a linear voice-coil actuator with double-wound, thin film heads and has an average access time of 15 milliseconds. The price is \$14,700 in sample quantities (no system interface). Reader Service No. 19.

Hitachi America Ltd.
950 Elm Ave.
San Bruno, CA 94066
(415) 872-1902

Diskit 2 Plus from **IDEAssociates** is an external hard-disk drive for the IBM PC that uses hardware encryption to protect the data on its removable 10-megabyte cartridges. The unit uses the Data Encryption Standard (DES) to prevent unauthorized access. It also offers on-line backup and an intelligent installation program that prompts users with plain-English questions. The price of \$3,595 includes software, controller, cabling, two 10-megabyte cartridges, and a maintenance kit. Reader Service No. 20.

IDEAssociates Inc.

29 Dunham Rd.
Billerica, MA 01821
(617) 663-6878

For the Macintosh

MORE is a high-end outline processor and organizational tool from **Living Videotext**. The integrated package includes full outlining and text processing, tree and bullet charts, cross-referencing, pattern matching, and many other features. The Macintosh version costs \$295 and is not copy-protected. Reader Service No. 21.

Living Videotext
2432 Charleston Rd.
Mountain View, CA 94043
(415) 964-6300

Hayes Microcomputer Products has introduced a communications device that connects AppleTalk networks locally or over modem connections. The product is called InterBridge and is equipped with AppleTalk and RS-232 ports. It sells for \$799. Reader Service No. 22.

Hayes Microcomputer Products Inc.
P.O. Box 105203
Atlanta, GA 30348
(404) 449-8791

Here's why you should choose Periscope as your debugger...

You'll get your programs running fast. "It works great! A problem we had for three weeks was solved in three hours," writes Wade Clark of MPPI, Ltd.

You'll make your programs solid. David Nanian says, "I can't live without it!! BRIEF, a text editor my company wrote, would not be as stable as it is today without Periscope."

You'll protect your investment. We won't forget you after the sale. You'll get regular software updates, including a FREE first update and notice of later updates. You'll get technical help from Periscope's author. And you'll be able to upgrade to more powerful models of Periscope if you need to. One Periscope user writes, "...

your support has won over even the heart of this hardened programmer!"

You deserve the best. Thousands of programmers rely on the only debugger that PC Tech Journal has ever selected as **Product of the Month** (1/86). You owe it to yourself to find out why, first hand.

You can try it at no risk. You get an unconditional 30-Day, Money-Back Guarantee, so you can't lose.

Start saving time and money now — order toll-free, 800/722-7006. Use MasterCard, Visa, COD, or a qualified company purchase order. As one user puts it, Periscope is "one of the rare products, worth every penny!"

Periscope I, software, manual, protected memory board and breakout switch	\$295
Periscope II, software, manual, and breakout switch	\$145
Periscope II-X, software and manual	\$115

Add shipping - \$3 US; \$8 Canada; \$24 elsewhere. Ask about air shipment if you can't wait to get your programs up and running!

PERISCOPE

by
Data Base Decisions, 14 Bonnie Lane,
Atlanta, GA 30328, 404/256-3860

The Macintosh can talk to Unix through MacNIX, a Macintosh icon-based interface to Unix systems. **Eurosoft International's** product includes a virtual file system that combines local Macintosh files and remote Unix files into one virtual directory tree. A host demon implements the interface on the Unix end while a Macintosh program runs locally. Installation can be done completely from the Macintosh. Prices range from \$2,000 to \$7,000 for the mainframe software. The local Macintosh disk costs \$49.95, and users are encouraged to copy it. Reader Service No. 23.

Eurosoft International Inc.
14082 Loma Rio Dr.
Saratoga, CA 95070
(408) 741-0739

MacBus is a hardware device from **National Instru-**

ments Corp. that lets a Macintosh Plus use IBM PC/AT interface cards. The unit has five AT-style slots, two of which are used for a microprocessor card and an interface card that connects to the SCSI port of the Mac Plus. The three remaining slots are available for any cards that are compatible with the AT bus. The microprocessor card contains a National Instruments GBIP-V50 and also supports an IEEE-488 interface. The hardware unit lists at \$1,495 and the Mac Plus software sells for \$200. Reader Service No. 24.

National Instruments Corp.
12109 Technology Blvd.
Austin, TX 78727
(800) 531-4742
in TX (800) IEEE-488

Hardware for the PC
The 386 Translator from American Computer and

Improve Program Quality Enhance Program Productivity

Design your programs around . . .

ASE, the Aspen Systems Subroutine Editor you can call from your programs. With **ASE** you can easily:

- ★ Design you own screen layouts for Program Input
- ★ Color and/or highlight input fields
- ★ Define your own key functions
- ★ Convert and edit a wide variety of fields
- ★ Update in several windows simultaneously

ASE includes 2 major subroutines, many minor subroutines and install and demonstration programs. Data & screen layouts described in a single map.

Price **\$99**

Available MSDOS 1.2, 3

Demo Disk **\$5**

ASP, the Aspen Systems Subroutine Package provides functions difficult or unavailable in some higher level languages, performs those available with greater speed/ease or smaller memory requirements.

The **ASP** Package includes:

- ★ 100+ subroutines
- ★ A 300 page manual packed with examples
- ★ Test, Demonstration and customization source

Price **\$130**

Available MSDOS 1.2, 3, CP/M

All subroutines are callable from assemblers and Microsoft Compilers, easily altered for other compilers, can be used in EXE or COM programs.

Prices are PPD (continental USA).

Colorado Residents add 3%.

P. O. Box 1163
Grand Junction, CO 81502
(303) 245-3262
VISA/MasterCard accepted

CP/M and MS-DOS are trademarks of Digital Research and Microsoft, respectively.

**ASPEN
SYSTEMS**

Circle no. 277 on reader service card.

*** PROGRAM WITH POWER ***

PC/POWER™

**Application Development
and Management System**

Runs on all 100% PC compatibles!

- Supports applications in a variety of languages
 - Including C, PASCAL, BASIC and Assembler
 - Even different languages in same application
- Screen painter - language and program independent screens
- Allows programs to pop-up menu/selection windows
- Builds indexes of applications and programs for easy use
- Supplies EXEC function to pass control between programs
 - One language can pass control to another
 - One language can even pass data to another
 - Inter-language considerations handled by PC/POWER
- Run-time supports control of multiple applications with:
 - Run-time start-up screen (customizable)
 - Pop-up application menu
- Integrates existing programs into an application
- Useful development applications included as samples

WITH RUNTIME MANAGER - NO ROYALTIES!

\$95.00 inc. Shipping and Handling

ORDERS & INQUIRIES (800) 628-2828 ext. 712

Beacon Street Software, Inc.

P.O. Box 216
Beacon Hill
Boston, MA 02133

ORDER NOW!



dBASE

to

C

the dBx™ translator

- **dBx** produces quality **C** direct from dBASE II or III programs.
- Move dBASE programs to UNIX or other machines.
- Improve program speed and reliability.
- Support multi-user/network applications.
- With power guidebook of conversion hints.
- Includes full screen handler and uses your current **C** database manager.
- May be used to move existing programs or help dBASE programmers learn **C** easily.
- For MSDOS, PC DOS, UNIX, XENIX, Macintosh, AMIGA. (Uses ANSI.SYS driver on MSDOS, CURSES under UNIX)
- Priced from \$350, also available from distributors.

dBx is a trademark of

Desktop Ai

1720 Post Road E., Westport, CT 06880
Phone • 203-255-3400

MCIMAIL • DESKTOPAI
Telex • 6502972226MCI

Circle no. 267 on reader service card.

HIRE A PRO!

DDJ announces a new service: Programmers' Opportunities


Dr. Dobb's Journal of Software Tools has a 10-year history of serving professional programmers with the most useful technical information of any publication for the computer industry. Now, in our new Programmers' Opportunities section, we are offering our readers information to help them stay on top of the quickly

changing technical-employment market.


We invite the top hardware, software, electronic, and instrument companies to list their computer-related career opportunities in *Dr. Dobb's Journal*. A limited number of companies will be given the chance to list jobs each month on a first-come, first-served basis.

For more information on how your company can hire a pro, write or call Gary George, 893 Monroe Dr., Atlanta, GA 30308; (404) 897-1923.


Our first responses are shown here as examples of what your listing will look like:



Position Title: 6502 assembly pgmr (C64/128)
Position Location: NY
Company/Agency: Bodylog, Inc.
Address: 34 Maple Ave.
City/State/Zip: Armonk, NY 10501
Phone: (914) 273-6480



Position Title: Sftwr Engr—Micro-based LANs
Position Location: suburban Chicago, Ill.
Company/Agency: United Airlines Dept EXOPX-RP
Address: P.O. Box 66100
City/State/Zip: Chicago, IL 60666
Phone: (312) 952-7329



Position Title: C Pgmr/Systems Analyst
Position Location: suburban Chicago, Ill.
Company/Agency: United Airlines Dept EXOPX-DF
Address: P.O. Box 66100
City/State/Zip: Chicago, IL 60666
Phone: (312) 952-7329

OF INTEREST

(continued from page 129)

Peripheral is a plug-in piggyback card that replaces the 80286 processor in an IBM PC/AT with an 80386. The device allows software designers to take advantage of the capabilities of the 80386 while still retaining full PC/AT compatibility. With an 80386 processor installed, the product costs \$895; without the CPU it costs \$395. Reader Service No. 25.
American Computer and Peripheral Inc.
2720 Croddy Way
Santa Ana, CA 92704
(714) 545-2004

Intel's Above Board PS/AT is an expanded-memory multifunction board for the IBM PC/AT. It complies with the Lotus/Intel/Microsoft Expanded Memory Specification and supports

up to 1.5 megabytes of memory alone or 3.5 megabytes with the Above Board Piggyback option. The board also supplies serial and parallel ports and several software utilities, including a RAM disk and print buffer. With 128K the board costs \$545; a 512K version sells for \$695. The Piggyback memory starts at \$295 for a 128K module. Reader Service No. 26.
Intel Corp.
5200 N.E. Elam Young Pkwy.
Hillsboro, OR 97124
(503) 629-7354

Alloy Computer Products' Bi-TURBO is a dual-tasking accelerator board that allows you to run two programs simultaneously by dedicating an on-board NEC V20 microprocessor to the second task. The board also contains 640K RAM for the second processor, 256K

disk cache RAM, and a private COM2 port for the second task. With software it costs \$995. Reader Service No. 27.
Alloy Computer Products Inc.
100 Pennsylvania Ave.
Framingham, MA 01701
(617) 875-6100

Networking

PC-Dial is a modem program from **ButtonWare** that runs on the IBM PC. Features include automatic log-on scripts of any length, DOS access, a mini-editor, and definable macro keys. It costs \$59.95. Reader Service No. 28.
ButtonWare
P.O. Box 5786
Bellevue, WA 98006
(206) 454-0479

Norton-Lambert's Close-Up is a communications package for the IBM PC that makes a remote PC into a

real-time window of a host PC. The package allows remote printing, graphics, file transfer, and full keyboard support. The host software costs \$245, and the slave software's price is set at \$195. Reader Service No. 29.
Norton-Lambert
P.O. Box 4085
Santa Barbara, CA 93140
(805) 687-8896

A communications program called **BackComm** from **LaSalle Micro** operates in the background on IBM PCs. It features keystroke learning for automatic connection, password protection, file encryption, and automatic call scheduling. It's priced at \$95. Reader Service No. 30.
LaSalle Micro Inc.
1350 Remington Rd., #W
Schaumburg, IL 60195
(312) 882-5171 x700

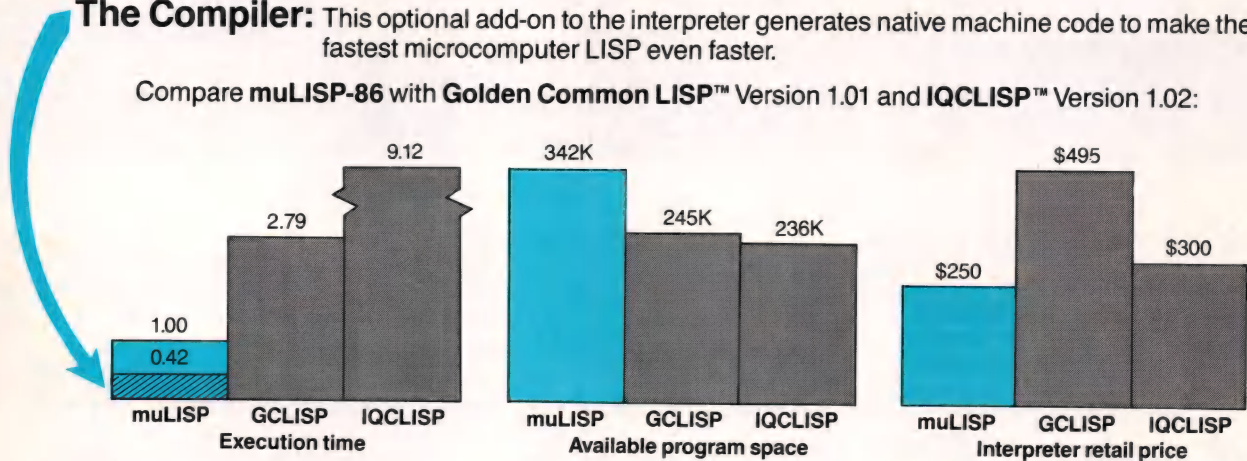
Introducing **muLISP-86™**

The professional AI programming environment for MS-DOS™ computers

The Interpreter: This powerful AI programming environment provides over 400 Common LISP functions, special forms, macros, and control variables.

The Compiler: This optional add-on to the interpreter generates native machine code to make the fastest microcomputer LISP even faster.

Compare **muLISP-86** with **Golden Common LISP™** Version 1.01 and **IQCLISP™** Version 1.02:



Write or call Soft Warehouse, Inc. for a more detailed comparison.

Execution time ratios based on the Gabriel Test Suite
Kilobytes of available program space on a 512K PC

MS-DOS is a registered trademark of Microsoft Corporation.
Golden Common LISP is a trademark of Gold Hill Computers.
IQCLISP is a trademark of Integral Quality.

Circle no. 168 on reader service card.



Soft Warehouse INC.

Founded 1979

3615 Harding Avenue, Suite 505 • Honolulu, Hawaii 96816
(808) 734-5801 after noon PST

A networking, multitasking OS called Waterloo Port for the IBM PC is now available in the U.S. from **Waterloo Microsystems**. It accommodates more than 150 PCs and runs MS-DOS as one of up to 12 simultaneous activities on each unit. The base price is \$1,695. Reader Service No. 31.

Waterloo Microsystems Inc.
175 Columbia St. W
Waterloo, ON N2L 5Z5
Canada

Languages

Mach 2 from **MicroHelp** is a collection of utilities for BASIC programmers using MS-DOS or PC-DOS. The utilities allow the programmer to break the 64K data limit of BASIC and incorporate a new method for including assembly-language routines in interpreted BASIC programs. Reader Service No. 32.

MicroHelp Inc.
2220 Carlyle Dr.
Marietta, GA 30062
(404) 973-9272

Chalcedony Software has released Prolog/m for the Macintosh and Prolog/i for the IBM PC. The languages support floating-point arithmetic, interactive debugging, and more than 100 predefined predicates and operators. They come with a built-in editor. The Macintosh version supports the full Mac interface. The IBM version supports the 8087 math chip and the large memory model. Prolog/m costs \$99.95, and Prolog/i costs \$69.95. Reader Service No. 33.

Chalcedony Software Inc.
5580 La Jolla Blvd.
La Jolla, CA 92037
(619) 483-8513

Software Development Systems has introduced

the UniWare 68020 Cross-Compiler System, which runs under Unix, Xenix, and MS-DOS. The package includes a C compiler, linker, librarian, and utilities. ROMable program images can be generated in several standard formats. The MS-DOS version costs \$595; Xenix and Unix versions cost \$1,390 and \$2,790, respectively. Reader Service No. 34.

Software Development Systems Inc.
3110 Woodcreek Dr.
Downer's Grove, IL 60515
(312) 971-8170

Microsoft has released a new BASIC compiler called QuickBASIC 2.0 for the IBM PC. The language offers high-speed, in-memory compilation and allows users to create structured and modular programs. It includes a built-in editor and debugger. It's priced at \$99.

Reader Service No. 35.
Microsoft Corp.
16011 N.E. 36th Way
Redmond, WA 98052
(206) 882-8080

Microsoft's Version 4.0 C compiler has several enhancements, including a new debugger called CodeView that uses windows to give programmers more complete control over the CPU and its environment. The Version 4.0 compiler also implements the Unix System V C library and supports the proposed ANSI standard. The new compiler, debugger, and library cost \$450. Reader Service No. 36.

Microsoft Corp.
16011 N.E. 36th Way
Redmond, WA 98052
(206) 882-8080

DDJ



**"Before I chose Microsoft C,
I spent 6 months evaluating C compilers
for my company. Now you can do
the same in 2 hours."**

*Bill Davidsen, Software Engineer
Office Automation Group
General Electric Research and Development*

**"Call us. You can get Microsoft C or
our comprehensive report on C by
the day after tomorrow."**

*Bruce Lynch, President
The Programmer's Shop*

The security of thorough research. It took Bill Davidsen six months to thoroughly evaluate all C products before he selected Microsoft C. For him, its tight code and UNIX System V™ compatibility were exactly what he needed. And now Version 4.00 includes CodeView™, a source-level windowing debugger.

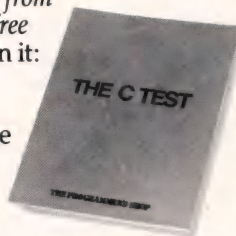
Thanks to expert users like Bill, and The Programmer's Shop, you can enjoy that satisfied feeling of thorough product evaluation in just a few hours.

We recommend evaluating software by also getting detailed information from several different sources, including unbiased reports and reviews. Bill agrees completely.

In fact, he helped us compile the objective opinions of 4 magazines, 14 users and 3 industry analysts in a 16-page report on C: *The C Test*. It can help you be absolutely sure of making the choice that's best for you. And it's absolutely free.

C for yourself. As an objective evaluation by users and professionals alike, *The C Test* is one of the most comprehensive and informative reports currently available on C development tools. *It's only available from The Programmer's Shop. And it's yours free for the asking.* Here's what you'll find in it:

- The C Test** ■ Detailed Tech Specs
■ Benchmark Source Code ■ Magazine Reviews
■ Users' Feedback ■ Performance Benchmarks
■ User Study and Profiles ■ Test Drive Survey Results
■ 37 Compatible Products



And if you're looking for even more C support, Microsoft-compatible libraries for file management, graphics, screen control, object-oriented programming and other tools are ready to ship.

The best programs for less. We think the only way to serve you is to give you the best programming alternatives. The best recommendations for your needs. To deliver immediately. And this is how we do it.

We start by giving you a choice of over 62 programming language implementations and 174 support programs. All from the same source. All competitively priced.

Our informed programmers offer free advice whenever you call with any questions about any product.

And when you place an order, we can rush it to you in 48 hours or less. That's the kind of service and support our 10,000 customers have come to expect.

Because we've become a success by giving the best advice for free and selling the best software for less.

To order Microsoft C (\$319) or for your free copy of *The C Test*, simply call the toll-free number below:

1-800-421-8006. In Massachusetts, call 1-800-442-8070.

MICROSOFT C Compiler Version 4.00

MICROSOFT C COMPILER

- Produces fast executables and optimized code including elimination of common sub-expressions. NEW!
- Implements register variables.
- Small, Medium and Large Memory model libraries.
- Compact and HUGE memory model libraries. NEW!
- Can mix models with NEAR, FAR and the new HUGE pointers.
- Library routines implement most of UNIX System V C library.
- Start-up source code to help create ROMable code. NEW!
- Full proposed ANSI C library support (except clock). NEW!
- Link your C routines with Microsoft FORTRAN (version 3.3 or higher), Microsoft Pascal (version 3.3 or higher) or Microsoft Macro Assembler.
- Microsoft Windows support and MS-DOS 3.1 networking support.

MICROSOFT PROGRAM MAINTENANCE UTILITY. NEW!

- Rebuilds your applications after your source files have changed.
- Supports macro definitions and inference rules.

OTHER UTILITIES.

- Library Manager.
- Overlay Linker.
- EXE File Compression Utility.
- EXE File Header Utility.

MICROSOFT CodeView

WINDOW-ORIENTED SOURCE-LEVEL DEBUGGER. NEW!

- Watch the values of your local and global variables and expressions as you debug.
- Set conditional breakpoints on variables, expressions or memory; trace and single step.
- Watch CPU registers and flags as you execute.
- Debug using your original source code, the resulting disassembly or both intermingled.

Microsoft C comes with a 30-day money-back guarantee from The Programmer's Shop.

UNIX System V is a trademark of AT&T Bell Laboratories.
Microsoft is a registered trademark and CodeView is a trademark of Microsoft Corporation.

THE PROGRAMMER'S SHOP

The programmer's complete source for software, services and answers.

128 Rockland Street, Hanover, MA 02339 **(617) 826-7531**

Circle no. 174 on reader service card.

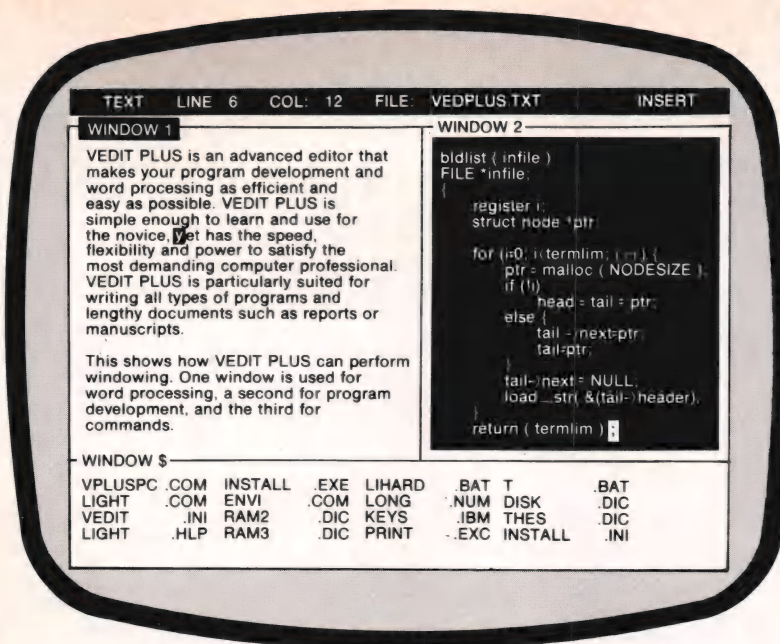
VEDIT PLUS

**MULTIPLE
WINDOWS**

**POP-UP
MENUS**

**KEYSTROKE
MACROS**

**EXECUTE DOS
PROGRAMS**



For over six years VEDIT has been the choice of professionals who demand the most powerful editing software available. CompuView has once again enhanced this power with the latest VEDIT PLUS - you can now open windows to simultaneously edit several files, access editing functions with pop-up menus, use keystroke macros to speed editing and run other programs within VEDIT PLUS.

Whether your needs are program development, technical writing or word processing, VEDIT PLUS is your answer. VEDIT PLUS is simple enough to learn for the novice, yet has the speed, flexibility and power to satisfy the most demanding computer professional. Its powerful macro programming language helps you eliminate repetitive editing tasks.

If you take your editing seriously, you need VEDIT PLUS. With over 40,000 users, you can depend on VEDIT PLUS to perform consistently and reliably. As have GE, EDS, U.S. Navy, GM, Sperry and many others. VEDIT PLUS supports color windows on the IBM CGA & EGA and even windows on most CRT terminals. Available for MS-DOS, PCDOS, CP/M-86 and CP/M-80. List price \$185.

"To sum things up, VEDIT PLUS is a small, fast, sophisticated editor with a wealth of features and a good macro language. It offers many rewards for the dedicated programmer."

Computer Language, Chris Wolf, Scott Lewis, Mark Gayman 6/86

"VEDIT PLUS is a wholly remarkable program: blindingly fast, extremely powerful, and highly flexible."

Profiles Magazine, Robert Lavenda 4/86

VEDIT PLUS FEATURES

- Simultaneously edit up to 37 files of unlimited size.
- Split the screen into variable sized windows.
- 'Virtual' disk buffering simplifies editing of large files.
- Memory management supports up to 640K.
- Execute DOS commands or other programs.
- MS-DOS pathname and CP/M user number support.
- Horizontal scrolling - edit long lines.
- Flexible 'cut and paste' with 36 text registers.
- Customization - determine your own keyboard layout, create your own editing functions, support any screen size, any CRT.
- Optimized for IBM PC/XT/AT. Also 132 column & up to 70 lines.

EASY TO USE

- Interactive on-line help is user changeable and expandable.
- On-line integer calculator (also algebraic expressions).
- Single key search and global or selective replace.
- Pop-up menus for easy access to many editing functions.
- Keystroke macros speed editing, 'hot keys' for menu functions.

FOR PROGRAMMERS

- Automatic Indent/Undent for 'C', PL/I or PASCAL.
- Match/check nested parentheses, i.e. '{' and '}' for 'C'.
- Automatic conversion to upper case for assembly language labels, opcodes, operands with comments unchanged.
- Optional 8080 to 8086 source code translator.

FOR WRITERS

- Word Wrap and paragraph formatting at adjustable margins.
- Right margin justification.
- Support foreign, graphic and special characters.
- Convert WordStar and mainframe files.
- Print any portion of file; separate printer margins.

MACRO PROGRAMMING LANGUAGE

- 'If-then-else', looping, testing, branching, user prompts keyboard input, 17 bit algebraic expressions, variables.
- CRT emulation within windows, Forms entry.
- Simplifies complex text processing, formatting, conversions and translations.
- Complete TECO capability.
- Free macros: • Full screen file compare/merge • Sort mailing lists • Print Formatter • Main menu

VEDIT and CompuView are registered trademarks of CompuView Products, Inc. MS-DOS is a registered trademark of Microsoft. CP/M is a registered trademark of Digital Research. WordStar is a registered trademark of MicroPro.

Circle no. 122 on reader service card.

CompuView

1955 Pauline Blvd., Ann Arbor, MI 48103 (313) 996-1299, TELEX 701821

COMBINE THE
RAW POWER OF FORTH
WITH THE CONVENIENCE
OF CONVENTIONAL LANGUAGES

HS/ FORTH

Why HS/FORTH? Not for speed alone, although it is twice as fast as other full memory Forths, with near assembly language performance when optimized. Not even because it gives MANY more functions per byte than any other Forth. Not because you can run all DOS commands plus COM and EXE programs from within HS/FORTH. Not because you can single step, trace, decompile & disassemble. Not for the complete syntax checking 8086/8087/80186 assembler & optimizer. Nor for the fast 9 digit software floating point or lightning 18 digit 8087 math pack. Not for the half megabyte LINEAR address space for quick access arrays. Not for complete music, sound effects & graphics support. Nor the efficient string functions. Not for unrivaled disk flexibility — including traditional Forth screens (sectored or in files) or free format files, all with full screen editors. Not even because I/O is as easy, but far more powerful, than even Basic. Just redirect the character input and/ or output stream anywhere — display, keyboard, printer or com port, file, or even a memory buffer. You could even transfer control of your entire computer to a terminal thousands of miles away with a simple >COM <COM pair. Even though a few of these reasons might be sufficient, the real reason is that we don't avoid the objections to Forth — WE ELIMINATE THEM!

Public domain products may be cheap; but your time isn't. Don't shortchange yourself. Use the best. Use it now!

HS/FORTH, complete system: \$395. with "FORTH: A Text & Reference" by Kelly and Spies, Prentice-Hall and "The HS/FORTH Supplement" by Kelly and Callahan



Visa

Mastercard



HARVARD SOFTWARES

PO BOX 69
SPRINGBORO, OH 45066
(513) 748-0390

Circle no. 132 on reader service card.

ADVERTISER INDEX

Reader Service No.	Advertiser	Page No.	Reader Service No.	Advertiser	Page No.
92	Addison Wesley	49	285	MDS, Inc.	53
178	Alloy Computer Products	21	95	MetaWare Incorporated	52
231	Apple Computer	50-51	*	Micro Way	33
121	Arity Corporation	118	215	MicroHelp, Inc.	46
248	Ashton Tate	135	*	Micromint	66
242	Ashton Tate	59	154	Microport Systems, Inc.	43
277	Aspen Systems	129	105	Microprocessors Unlimited	124
250	Austin Code Works	82	*	Microtec Research	99
115	Barrington Systems, Inc.	4-5	156	Mix Software	127
182	BC Associates	99	249	Mortice Kern Systems, Inc.	125
267	Beacon Street Software, Inc.	129	*	Motorola Inc.	26-27
159	Blaise Computing	121	*	Nanosoft Associates	119
217	Blaise Computing	120	220	Nantucket Corporation	65
161	Borland International	C4	243	Norton Utilities	125
212	Burton Systems Software	114	251	Nostradamus	2
181	C Users Group	124	227	Oakland Group, Inc.	72
226	Cauzin Systems	116-117	254	Oasys	113
150	Chalcedony Software, Inc.	88	224	Personal Computer Support Group	C3
122	Compu View	133	76	Personal Tex	95
237	CompuServe	15	239	PMI	99
94	Consulair Corporation	78	283	Polytron Corporation	61
238	Cracker Jack Microsoft Corp.	82	129	Programmer's Connection	68-69
*	Creative Programming	37	98	Programmer's Connection	71
268	Custom Software Systems	95	103	Programmer's Connection	73
214	Data Based Decisions	128	86	Programmer's Connection	75
203	Datalight	9	174	Programmer's Shop	132
*	Desktop A.I.	129	143	Programmer's Shop	87
87	Digital Research Computers	103	141/133	Programmer's Shop	92-93
*	DDJ Allen Holub-Shell	112	295	Proto PC	95
*	DDJ Bound Volumes	106-107	206	Raima Corporation	23
*	DDJ Code Listings	110	145	Rational Systems	70
*	DDJ C-Products	108-109	*	SAS Institute	47
*	DDJ Toolbook of Forth	111	183	SBC Mart	83
*	DDJ Z80 Toolbook	111	210	Scientific Endeavors	14
89	Ecosoft, Inc.	98	85	Semi Disk Systems	55
*	Edward K. Ream	88	78	SLR Systems	88
173	Entelekon	77	83	Soft Advances	76
138	Essential Software	74	168	Soft Warehouse, Inc.	131
93	Fair-Com	98	113	Softcraft Inc.	19
216	Forth Interest Group	119	259	Softfocus	94
*	Gimpel Software	63	218	Software Directions, Inc.	41
97	Greenleaf Software	67	170	Software Security, Inc.	37
132	Harvard Softworks	134	142	Solution Systems	91
274	Hauptpauge Computer Works	13	148	Solution Systems	97
176	Haventree Software	100	152	Solution Systems	97
233	Hawaiian Village Computers	35	185	Tall Tree Systems	C2
194	Info Pro Systems	101	279/245	Tech PC	80-81
*	Integral Quality, Inc.	90	230	TSF	76
179	Intel Corporation	30-31	207	Turbo Power Software	89
190	Intel Corporation	79	119	Turbo Tech Report DDJ Newsletter	17
205	Laboratory Microsystems, Inc.	101	157	Vermont Creative Software	18
186	Lahey Computer Systems, Inc.	101	189	Warp Speed Light Pen	71
266	Language Processors, Inc.	71	112	Wendin	11
101	Lattice, Inc.	25	211	Western Computer	57
118	Lifeboat/Scarborough Systems	119	116	Wizard Systems	102
257	Logitech, Inc.	123	208	Wordtech Systems	115
135	Lugaru	122	244	Workman & Associates	86
187	MacTutor	86	225	Xenosoft	114
108	Manx Software Systems	7	162	Z-World	126
102	Mark Williams Company	1			

*This advertiser prefers to be contacted directly: see ad for phone number.

ADVERTISING SALES OFFICES

Midwest

Michele Beaty (317) 875-8093

Southeast

Gary George (404) 897-1923

Northeast

Cynthia Zuck (718) 499-9333

Northern California/Northwest

Lisa Boudreau (415) 366-3600

Southern California/AZ/NM/TX

Michael Wiener (415) 366-3600

Advertising Director

Robert Horton (415) 366-3600

dBASE to the max.

Introducing a turbocharger for dBASE®.

Now the world's leading dBASE gurus, Jeb Long, Robert Byers and Wayne Ratliff, have combined their own personal utilities into a collection of dBASE programming tools that'll allow you to maximize your output while minimizing your time and effort.

Included are "secrets of the stars" on dBASE database repair and recovery, program structuring and cross referencing for dBASE programs, restricting access to a dBASE database, communications port control from within dBASE, menubar creation, saving and restoring screens, and much more.

It's "insider information" you can't go to jail for using.

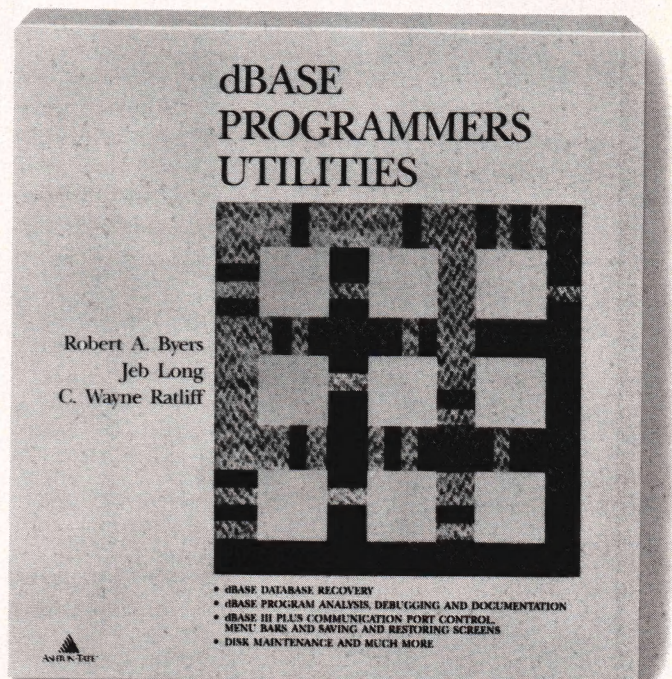
For merely \$89.95 you have at your fingertips what the database honchos do: over a dozen DOS level programs to assist you with disk and directory management, time displays, and rapid

searching for file names and switching between screen buffers.

Not to mention plenty of other neat stuff.

To order, or for more information or the name of your nearest dealer, call the Ashton-Tate® Publishing Group at (800) 437-4329, Ext. 221.

And put your dBASE into high gear.



SWAINE'S FLAMES

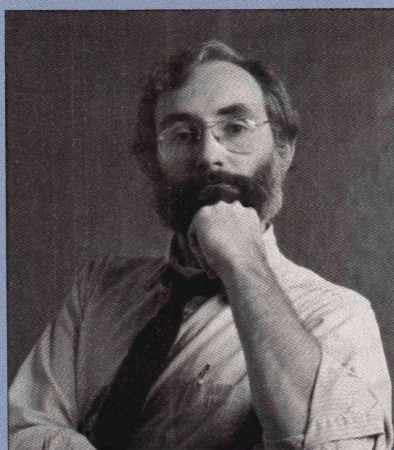
The question is, as Humpty Dumpty said to Alice, Which is to be master?

Andy Hertzfeld wrote the Macintosh Finder as an Apple employee. He's no longer an Apple employee, but he hasn't lost his fondness for the computer and hasn't quit tweaking its operating system. His Switcher has become for many users of the Mac an essential component of the system software, making the difference between feeling like the machine's master and feeling like the machine's the master.

And now there's Servant, Hertzfeld's planned replacement for Finder. Servant will allow you to keep several applications open at a time and move easily between applications and the desktop, may support batch-file operations, and should generally speed up the operations that tend to be slow on the Mac. Sounds like what the doctor ordered, but what if (just what if) Apple should decide not to distribute Servant and Hertzfeld decided to do so? Could Hertzfeld simply take the product to the users? Will the Macintosh come when Andy Hertzfeld whistles?

Phoenix Technologies would like someone other than IBM to be the master of the 80386. Since the spring Comdex, Phoenix has been trying to get companies developing 80386-based machines to agree to a bus standard for the machines. Companies other than IBM, that is. Whether or not Phoenix gets cooperation, it looks like we may have a choice of "standard" 80386 architectures.

The real which-is-to-be-master question for the 80386 concerns the operating system, with most analysts insisting that, after the initial phase in which the 80386 merely will serve to make PC-DOS 3.x run faster, one operating system must emerge as the winner. Microsoft is only now alpha-testing its DOS 5.0 (which supports protected mode for the 80286) and



probably won't release DOS 6.0 for the 80386 before the end of 1987. AT&T, on the other hand, has Unix System V/386 in beta, and it's scheduled for release in November, so we can expect to see compilers that run under Unix.

In fact, a bundle recently arrived from Regis McKenna announcing various artificial intelligence products for the 80386 that supposedly are coming out in the next six months. Gold Hill, Lucid, and Franz are all producing Common LISP compilers, interpreters, and development tools. Franz is also porting Flavors, its object-oriented programming environment, to the 386. And Arity is bringing its version of PROLOG over. All the products have foreign-language interfaces for C and other languages, and all are targeted for Unix System V/386.

It makes sense to use the 80386 for AI work. Beyond the fact that AI work simply needs a lot of processing power, AI programs typically jump all over the map and benefit from the flat memory space of a processor like the 80386.

Meanwhile, IBM has decided to distribute a version of LISP (Lucid's) on its 32-bit RISC-architecture RT/PC. The operating system of the RT/PC is a Unix System V derivative.

Of course, not everyone loves Unix. Ken Williams of Softguard Systems is traveling around talking to programmers about VM386, the multitasking operating system his company is developing for the 80386 that is intended to support DOS applications while

taking advantage of the advanced features of the 80386.

Then we have Hunter Systems trying to put DOS on 68000 machines. Hunter argues that if DOS were specified in C, like Unix, it would be as portable as Unix. The company has been working with several hardware and software vendors (including Motorola) to develop a portable DOS, including ROM BIOS and video RAM, all written in C. The DOS would be compatible with Microsoft's DOS 5.0 as well as existing DOS versions.

In the August Flames I wrote of my cousin Corbett's proposal that all PROLOG programmers adopt a uniform commenting style so that some future compiler directive could turn the comments into references to a universal dictionary of PROLOG-style facts, the notion being that the comprehension of comments would rely on human intelligence only until something better comes along. This prompted Stan Kelly-Bootle, author of the wonderful *Devil's DP Dictionary*, to remind me that he presented a related idea in his Devil's Advocate column in February's *Unix Review*. Stan's yacc (yet another comment compiler) would be a great boon in converting programs written in some soon-to-be-obsolete language like, say, C, to powerful AI code. It would ignore code and interpret comments, turning the pedestrian

`++a /* increment count by 1 */`

into the soaring

`increment count by 1 /* ++a */`

with all the obvious resultant benefits.

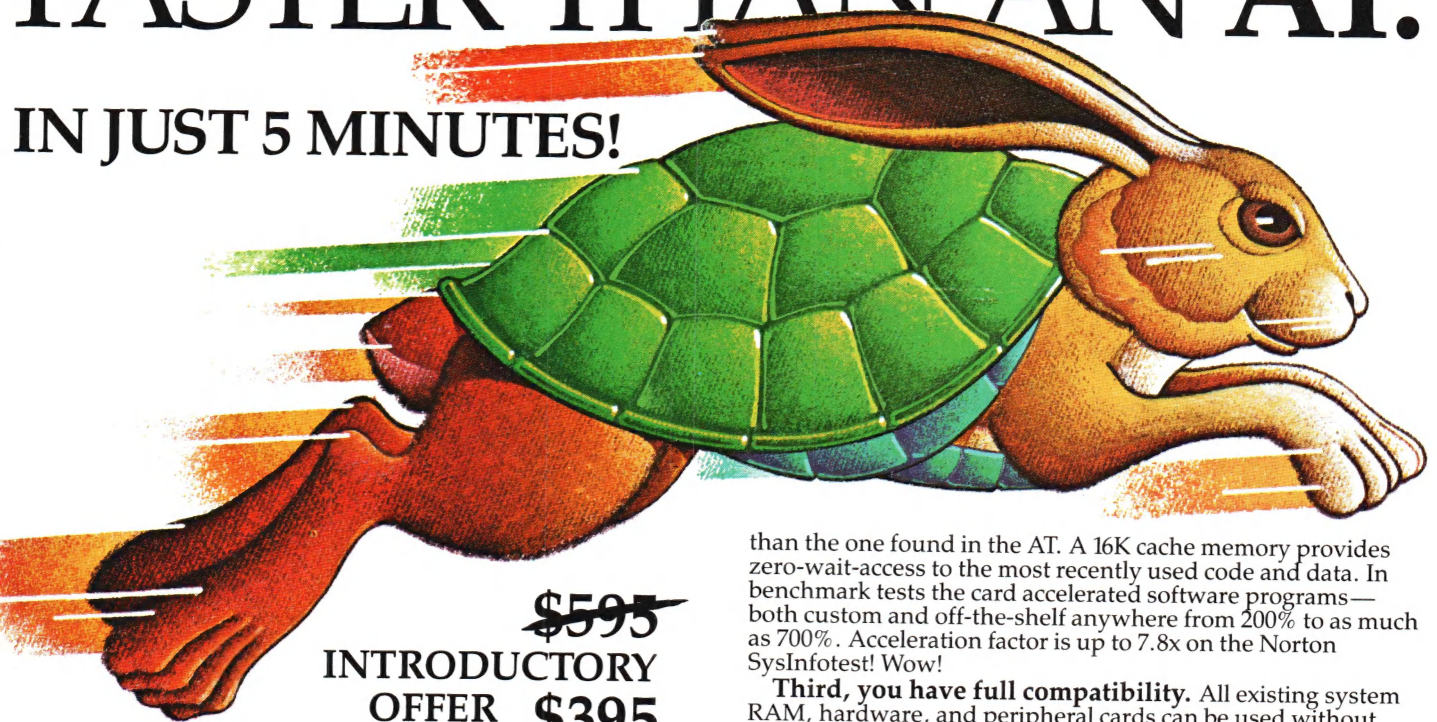
As Humpty Dumpty said, there's glory for you.

Michael Swaine

Michael Swaine
editor-in-chief

MAKE YOUR IBM PC FASTER THAN AN AT.

IN JUST 5 MINUTES!



~~\$595~~
**INTRODUCTORY
OFFER \$395**

**DON'T TAKE OUR WORD FOR IT.
USE IT FOR 60 DAYS. IF YOU ARE
NOT TOTALLY SATISFIED RETURN
IT FOR A FULL REFUND.**

It sounds great; the idea of a speedup board that you can just plug right in as easily as putting bread in a toaster. How wonderful to be able to convert a PC or XT to a \$4000 AT without the expense. But even when you get ready to spend \$595.00 you want to be sure your choice is the very best.

Here at PCSG we sell our IBM PC disk access speedup software by the thousands. But software doesn't do anything about speeding up the microprocessor (or CPU) speed. As you know the microprocessor is the brain of the computer that controls all the operations like screen updates and calculations like a spreadsheet makes.

***Faster and smarter than an AT—
PCSG guarantees it.***

We wanted to offer a speedup card that would be the complement to our disk speedup software, (incidentally included at no extra charge.) We wanted it to be literally the most advanced, compatible and feature rich board available today. We could only be satisfied with a board that was the finest example of the engineering art.

There is no question we have met our every objective by developing and manufacturing the *BREAKTHRU 286* card. This is the best designed and most functional speed up card available today. We guarantee it.

HERE IS WHAT MAKES IT SO SPECIAL.

First, it installs so easily. It is a half slot card, only five inches in length. You don't even have to give up a full slot. What's more, unlike competing products it works in the Compaq and most clones. The instructions are so simple we considered showing a picture of a child putting it in. Easy diagrams show how you just place the card in an open slot, remove the original processor and connect a single cable. There is no software required. From that moment you are running faster than an AT.

Second, it is advanced. The *BREAKTHRU 286* replaces the CPU of the PC or XT with an 80286 microprocessor that is faster

than the one found in the AT. A 16K cache memory provides zero-wait-access to the most recently used code and data. In benchmark tests the card accelerated software programs—both custom and off-the-shelf anywhere from 200% to as much as 700%. Acceleration factor is up to 7.8x on the Norton SysInfoTest! Wow!

Third, you have full compatibility. All existing system RAM, hardware, and peripheral cards can be used without software modification. It operates with LAN and mainframe communication products and conforms to the Lotus/Intel/Microsoft Expanded Memory Specification (EMS). Software compatibility is virtually universal.

Fourth, it is the best there is. There are several other boards on the market. Some are priced about the same as the *BREAKTHRU 286* and some are cheaper. We at PCSG have compared them all, but there simply was no comparison. What we discovered is that many cards being sold offer only a marginal speed up in spite of their claims. We found some to be merely versions of the obsolete 8088 or 8086, and others to be just poorly engineered. The 8MHz *BREAKTHRU 286* is unequivocally the best executed and most completely reliable speedup board manufactured today.

PCSG has since early 1983 dominated the lap portable market with ROM software such as Lucid spreadsheet and Write ROM that reviewers rated as excellent. We were proud to successfully enter the IBM PC market last year with disk access speedup software. Now we are so pleased with the *BREAKTHRU* speedup card. We use them on our own PC's to make them faster than AT's. We are really excited about this product.

PCSG makes the unabashed statement that the *BREAKTHRU 286* card represents more advanced technology than boards by Orchid, Quadram, Victor, Mountain, P.C. Technologies, Phoenix... we could go on.

But an ad can't let you experience it for yourself. That's why we sell the *BREAKTHRU 286* on a 60 day trial. If you aren't completely satisfied return it within 60 days for a full refund. It is priced at \$595. Call today with your MasterCard, Visa, American Express or COD instructions and we will ship your card the very next day.

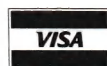
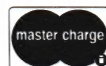


Circle no. 224 on
reader service card.

PERSONAL COMPUTER SUPPORT GROUP

11035 Harry Hines Blvd. #207 • Dallas, Texas 75229

214-351-0564



Step-by-step tutorial, demo programs with source code included!

Borland's new Turbo Prolog is the powerful, completely natural introduction to Artificial Intelligence

Prolog is probably one of the most powerful computer programming languages ever conceived, which is why we've made it our *second* language—and "turbocharged" it to create Turbo Prolog.*

Our new Turbo Prolog, the natural language of **Artificial Intelligence**, brings supercomputer power to your IBM® PC and introduces you step-by-step to the fascinating new world of **Artificial Intelligence**. And does all this for an astounding \$99.95.

Turbo Prolog is to Prolog what Turbo Pascal® is to Pascal!

Our Turbo Pascal astonished everyone who thought of Pascal as "just another language." We changed all that—and now Turbo Pascal is the de facto worldwide standard, with hundreds of

thousands of enthusiasts and users in universities, research centers, schools, and with professional programmers, students, and hobbyists.

You can expect at least the same impact from Turbo Prolog, because while Turbo Prolog is the most revolutionary and natural programming language, it is also a complete development environment—just like Turbo Pascal.

"Turbo Prolog offers generally the fastest and most approachable implementation of Prolog."

Darryl Rubin,
AT Expert

Even if you've never programmed before, our free tutorial will get you started right away

You'll get started right away because we have included a complete step-by-step tutorial as part of the 200-page Turbo Prolog Reference Manual. Our tutorial will take you by the hand and teach you everything you're likely to need to know about Turbo Prolog and artificial intelligence.

For example: once you've completed the tutorial, you'll be able to design your own expert systems utilizing Turbo Prolog's powerful problem-solving capabilities.

Think of Turbo Prolog as a high-speed electronic detective. First you feed it information and teach it rules. Then Turbo Prolog "thinks" the problem through and comes up with all the reasonable answers—almost instantly.

If you think that this is amazing, you just need to remember that Turbo Prolog is a 5th-generation language—and the kind of language that 21st century computers will use routinely. In fact, you can compare Turbo Prolog to

Turbo Pascal the way you could compare Turbo Pascal to machine language.

You get the complete Turbo Prolog programming system for only \$99.95

You get a complete Turbo Prolog development system including:

- The lightning-fast Turbo Prolog incremental compiler and the interactive Turbo Prolog editor.
- The 200-page reference manual which includes the step-by-step Turbo Prolog tutorial.
- The free GeoBase™ natural query language database including commented source code on disk—ready to compile. GeoBase is a complete database designed and developed around U.S. geography. It includes cities, mountains, rivers, and highways, and comes complete with natural query language. Use GeoBase immediately "as is," or modify it to fit your own interests.

So don't delay—don't waste a second—get Turbo Prolog now. \$99.95 is an amazingly small price to pay to become an immediate authority—an instant expert on artificial intelligence! The 21st century is only one phone call away.

Turbo Prolog 1.0 Technical Specifications Programming System Features

- ✓ **Compiler:** Incremental compiler generating native in-line code and linkable object modules. The linking format includes a linker and is compatible with the PC-DOS linker. Large memory model support. Compiles over 2500 lines per minute on a standard IBM PC.
- ✓ **Interactive Editor:** The system includes a powerful interactive full-screen text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code. At run-time, Turbo Prolog programs can call the editor, and view the running program's source code.
- ✓ **Type System:** A flexible object-oriented type system is supported.
- ✓ **Windowing Support:** The system supports both graphic and text windows.
- ✓ **Input/Output:** Full I/O facilities, including formatted I/O, streams, and random access files.
- ✓ **Numeric Ranges:** Integers: -32767 to 32767; Reals: 1E-307 to 1E+308
- ✓ **Debugging:** Complete built-in trace debugging capabilities allowing single stepping of programs.

YES!

I want the best

Turbo Prolog at only:

\$99.95

To order by phone,
or for a dealer nearest you,
Call (800) 255-8008
in CA call (800) 742-1133.

Send me ☐ Turbo Prolog at \$ _____

Outside USA add \$10 per copy

CA and MA res. add applicable sales tax \$ _____

Amount enclosed: \$ _____

This price includes shipping to all US cities

Payment: ☐ VISA ☐ MC ☐ Bank Draft ☐ Check

Credit card expiration date: ____/____/____

Card #

 **BORLAND**
INTERNATIONAL

Vive la différence

4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

Borland products include Turbo Prolog, Turbo Pascal, Turbo Tutor, Turbo Editor Toolbox, Turbo Database Toolbox, Turbo Graphics Toolbox, Turbo GameWorks, Turbo Lighting, Lightning Word Wizard, Reflex, The Analyst, Reflex Workshop, Sidekick, Sidekick, The Macintosh Office Manager, Traveling Sidekick, and SuperKey—all of which are trademarks or registered trademarks of Borland International, Inc. or Borland Analytics, Inc.

Copyright 1986 Borland International, Inc. BI-1045E

IBM, XT, AT, and PCjr are registered trademarks of International Business Machines Corp.